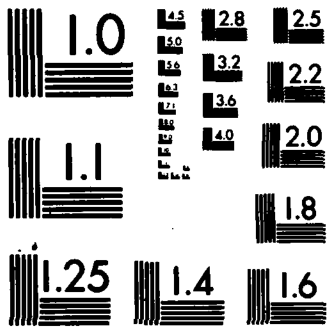
AD-A140 268 REVISIONS TO THE PETROS 4 SHELL RESPONSE CODE(U) ARMY ARMAMENT RESEARCH AND DEVELOPMENT CENTER ABERDEEN PROVIN.. N J HUFFINGTON ET AL. FEB 84 ARBRL-TR-02550 1/1

UNCLASSIFIED SBI-AD-F300 408 F/G 1/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-F300408

(11)

AD

ADA140268

TECHNICAL REPORT ARBRL-TR-02550

REVISIONS TO THE PETROS 4 SHELL RESPONSE CODE

Norris J. Huffington, Jr.
Henry L. Wisniewski

February 1984

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT CENTER
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

84 04 16 058

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: Technical Report ARBRL-TR-02550	2. GOVT ACCESSION NO.: AD-A140 268	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Revisions to the PETROS 4 Shell Response Code		5. TYPE OF REPORT & PERIOD COVERED: Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Norris J. Huffington, Jr.; Henry L. Wisniewski		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: US Army Ballistic Research Laboratory, ARDC; ATTN: DRSMC-BLV-V(A); Aberdeen Proving Ground, MD 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 1L161102AH43
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army AMCCOM, ARDC; Ballistic Research Laboratory, ATTN: DRSMC-BLA-S(A); Aberdeen Proving Ground, MD 21005		12. REPORT DATE: February 1984
		13. NUMBER OF PAGES: 93
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report): UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Structural Panels
Air Blast
Elastoplastic Material
Transverse Shear Deformation
Surface Traction Effects
Finite Difference Method

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The existing PETROS 4 finite deflection elastoplastic structural shell response code was modified to permit analysis of the response of structural panels to nearby high explosive detonations. A modified constitutive formulation involving prescribed through-thickness stresses was introduced. Previously omitted surface traction terms were added to the equations of motion. A recycling option based on a strain equivalence criterion was introduced to inhibit the unrestricted growth of a breathing mode. Additional printer and plotter output features were incorporated into the program.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

LIST OF ILLUSTRATIONS

I. INTRODUCTION

A recurring problem in vulnerability analysis is the requirement to predict the damage to structural panels produced by the detonation of high explosive charges in close proximity to the target panels. Examples of problems where this requirement arises are (a) the blast effect of shaped charge warheads against lightly armored vehicles and (b) the effect of small caliber high explosive shells which detonate close to aircraft panels. Although both these examples involve additional damaging effects (e.g., the shaped charge jet, and shell casing fragments), this report is concerned only with a rational analysis of the response induced by the air blast, leaving the combined effects of all lethal mechanisms for subsequent consideration.

While the methodology for blast loading prediction is far from satisfactory at present, in the sequel it will be assumed that the blast pressure on the target surface is a known function $p\ (r, \theta, t)$.* Since we will be concerned with the blast from conventional chemical explosives the pressure will be of high intensity and short duration, resulting in the delivery of a significant impulse to the target. However, the peak pressure will be assumed insufficient to produce spallation from the back side of the target panel, thus excluding consideration of explosives in contact with the target. This is not a serious restriction, since stand-off explosions can produce catastrophic damage to structural panels.

It is now necessary to adopt a structural response analysis tool which has attributes sufficient for adequate modeling of the physical phenomena expected to occur. Specifically the analysis should be capable of treating finite amplitude elastoplastic response of shell structures having a variety of physical edge conditions, which are subjected to transient distributions of surface pressures. It should have flexibility regarding material constitutive representation, including strain hardening and strain-rate dependence. It should also be possible to introduce various material failure models into this analysis and, preferably, to perform some post-failure calculations. Further, since the blast load will initially appear at some interior point on the panel and then spread rapidly to cover the entire panel, the analysis tool should properly account for the propagation of shear waves as well as flexural and membrane waves. Finally, it is desired to avoid a general three-dimensional analysis, if possible, for reasons of computational economy.

Taking the foregoing considerations into account, a variety of available finite element and finite difference computer programs were reviewed and it was decided that the PETROS 4 code[1,2] provided the best point of departure for meeting these requirements.

---

* *See Nomenclature, p. 41, for definition of symbols.*

[1] *S. D. Pirotin, L. Morino, E. A. Witmer, and J. W. Leech, "Finite-Difference Analysis for Predicting Large Elastic-Plastic Transient Deformations of Variable-Thickness Kirchhoff, Soft Bonded Thin, and Transverse-Shear Deformable Thicker Shells," US Army Ballistic Research Laboratory Contract Report No. 315, September 1976.* AD B013924L

[2] *S. D. Pirotin, B. A. Berg, and E. A. Witmer, "PETROS 4: New Developments and Program Manual for the Finite-Difference Calculation of Large Elastic-Plastic, and/or Viscoelastic Transient Deformations of Multilayer Variable-Thickness (1) Thin Hard-Bonded, (2) Moderately-Thick Hard-Bonded, or (3) Thin Soft-Bonded Shells," US Army Ballistic Research Laboratory Contract Report No. 316, September 1976.*

This report is concerned with documentation of the modifications to this code which were required to achieve the desired analysis capability.

II. DEFICIENCIES OBSERVED IN THE ORIGINAL PETROS 4 PROGRAM

The user of the PETROS 4 code has first to choose one of the three versions referred to in the title of Reference 2. For the reasons previously stated the primary attention has been given to the moderately-thick hard-bonded* transverse shear deformable option although occasional use of the thin hard-bonded Kirchhoff shell model has been made for comparison purposes. Another choice to be made is the plasticity theory to be employed, which is selected by the value assigned to the input variable ISTRES, as follows:

ISTRES	Plasticity Theory
0	Mechanical sublayer model[3,4,5], 3-D stress
1	Prandtl-Reuss model, 3-D stress
2	Mechanical sublayer model, $\tau^{3j} = 0$ $(j=1,2,3)$ This option was recommended for the Kirchhoff shell.
3	Mechanical sublayer model, $\tau^{33} = 0$ (This stress component is oriented along the normal to the shell reference surface.)

---

* *The PETROS 4 code can treat shells composed of layers of different materials (although the immediate interest is in applications involving only a single layer).*

3 *H. F. Bohnenblust, and P. Duwez, "Some Properties of a Mechanical Model of Plasticity," Journal of Applied Mechanics, Vol. 15, No. 3, September 1948, pp. 222-225.*

4 *G. N. White, Jr., "Application of the Theory of Perfectly Plastic Solids to Stress Analysis of Strain Hardening Solid," Graduate Div. of Applied Math., Brown University Tech Report 51, August 1950.*

5 *J. F. Besseling, "A Theory of Plastic Flow for Anisotropic Hardening in Plastic Deformation of an Initially Isotropic Material," Report 5410, National Aeronautical Research Institute, Amsterdam, The Netherlands, 1953.*

Another input quality which must be selected is INORML, which controls the manner in which the variable D^3 is calculated. D^3 is the component of the vector $\overline{D}$ in the direction of the normal to the reference surface and $\overline{D}$ represent the three additional degrees-of-freedom of the SHEAR model besides those of the Kirchhoff model at each mesh point. The options for INORML are:

INORML	$\overline{D}$ Calculation
0	The cartesian components of $\overline{D}$ are calculated using three equations of motion
1	D^3 is set to zero after $\overline{D}$ is calculated
2	The incremental change in D^3 is calculated from the incremental strain $\Delta\gamma_3^3$ using the elastoplastic constitutive relations; this corresponds to a thickness change which affects the stresses at the next time step.

A. Stress Calculation Inconsistencies

In order to treat the problem of a blast-loaded structural plate it would appear appropriate to use the SHEAR *version of the* PETROS 4 *code* with the options ISTRES = 0, INORML = 0 since these are the most general (least restrictive) choices available. This combination of options has been employed to treat the following physical example:

A square plate of rolled homogeneous steel armor, 0.1905m (7.50 in) by 0.1905m (7.50 in) by 9.53mm (0.375 in) thickness. Young's modulus: 2.068GPa (30×10^6 psi) Poisson's ratio: 0.25. The uniaxial strain-hardening characteristics of this material were represented in the mechanical sublayer model by the following stress-strain coordinates (connected by linear segments):

Coordinate No.	Stress	Strain
1	1.048 GPa (152000 psi)	0.005067
2	1.145 GPa (166000 psi)	0.0135
3	1.248 GPa (181000 psi)	0.0530
4	1.675 GPa (243000 psi)	0.2800

The boundary conditions imposed on all four edges were complete fixity. The plate was loaded by the blast from a 0.907kg (2 pound) spherical pentolite charge detonated 63.5mm (2.5 in) above its midpoint.

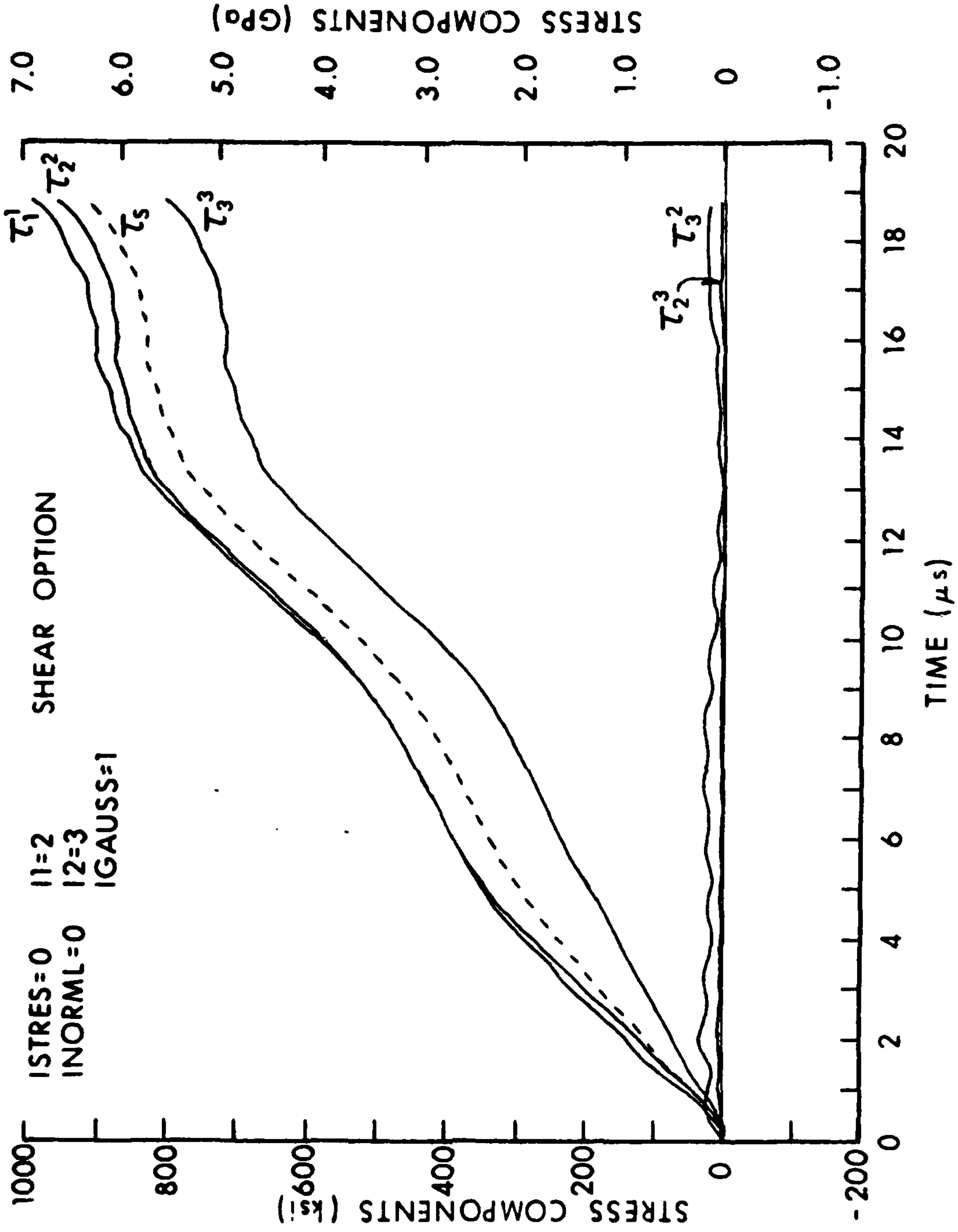


Figure 1. Transient Stress Components for Indicated Options

Results of PETROS 4 calculations of transient stress components using the cited options are shown in Figure 1. The location of these stresses is a point 9.53mm (0.375 in) from the plate midpoint along a line perpendicular to an edge and 0.661mm (0.026 in) above the lower surface. One sees that all three normal stress components have become very large (as has the mean or hydrostatic stress τ_s) by the end of this short run and it may be readily verified that the elastoplastic stress components satisfy the von Mises yield function and the associated flow rule. On the other hand the prescribed blast overpressure at this location jumps to a peak value of 834MPa (120900 psi) and decays exponentially to 621MPa (90000 psi) by the end of the run (while the pressure on the lower surface remains at zero). Consequently, one would expect that the mean value of the through-thickness stress component τ_3^3 would be negative (compressive) and that any tensile excursion would be small. For this reason it is felt that the normal stresses displayed in Figure 1 are exceedingly suspect.

Before speculating on the cause of this behavior let us compare solutions of the same physical problem obtained by use of other options of the PETROS 4 code. Using the Kirchhoff model with ISTRES = 2, INORML = 0 the distinctly different and more plausible results shown in Figure 2 were derived. Rather than a runaway increase, the normal stresses τ_1^1 and τ_2^2 appear to be reaching a maximum at stress levels which armor plate may sustain. However, this Kirchhoff solution has the drawbacks that (1) a two-dimensional constitutive relation is employed ($\tau_3^3 \equiv 0$) so that the boundary condition on the upper surface cannot be satisfied and (2) the early time solution may be inaccurate since transverse shear deformation is neglected.

Now consider solutions of the same problem derived using the SHEAR model with ISTRES = 3 where the three-dimensional constitutive relation is utilized subject to the constraint $\tau_3^3 \equiv 0$. For INORML = 0 the predicted stresses are displayed in Figure 3. These stresses appear entirely plausible; however, experience with longer computer runs using this option combination has revealed a tendency to unchecked growth in magnitude of the variable D^3 and the associated through-thickness strain component γ_3^3. For INORML = 1 the calculated stresses are plotted in Figure 4. These results, while different from the preceding, are also plausible. In this case transverse shear deformation is permitted but the "breathing" deformation mode is inhibited by the non- physical constraint $\gamma_3^3 \equiv 0$. Finally, the solution for INORML = 2 is presented in Figure 5. It had been expected that this option combination would provide the "best" predictions since both shear deformation and "breathing" are permitted. However, in this and all other runs made with this option unstable results, including negative plastic work, were obtained. It must be concluded that there exists an error in either the formulation or the coding for this option.

It is apparent that each option combination leads to a different solution for the stresses. The solutions shown in Figs. 1 and 5 are obviously unsatisfactory. The Kirchhoff shell analysis of Figure 2 is correct within the limitations of classical thin shell theory but a more refined analysis including transverse shear deformation is desired. The differences between the results of Figures 3 and 4 are significant; in sequel these differences will be explained and a more satisfactory formulation derived.

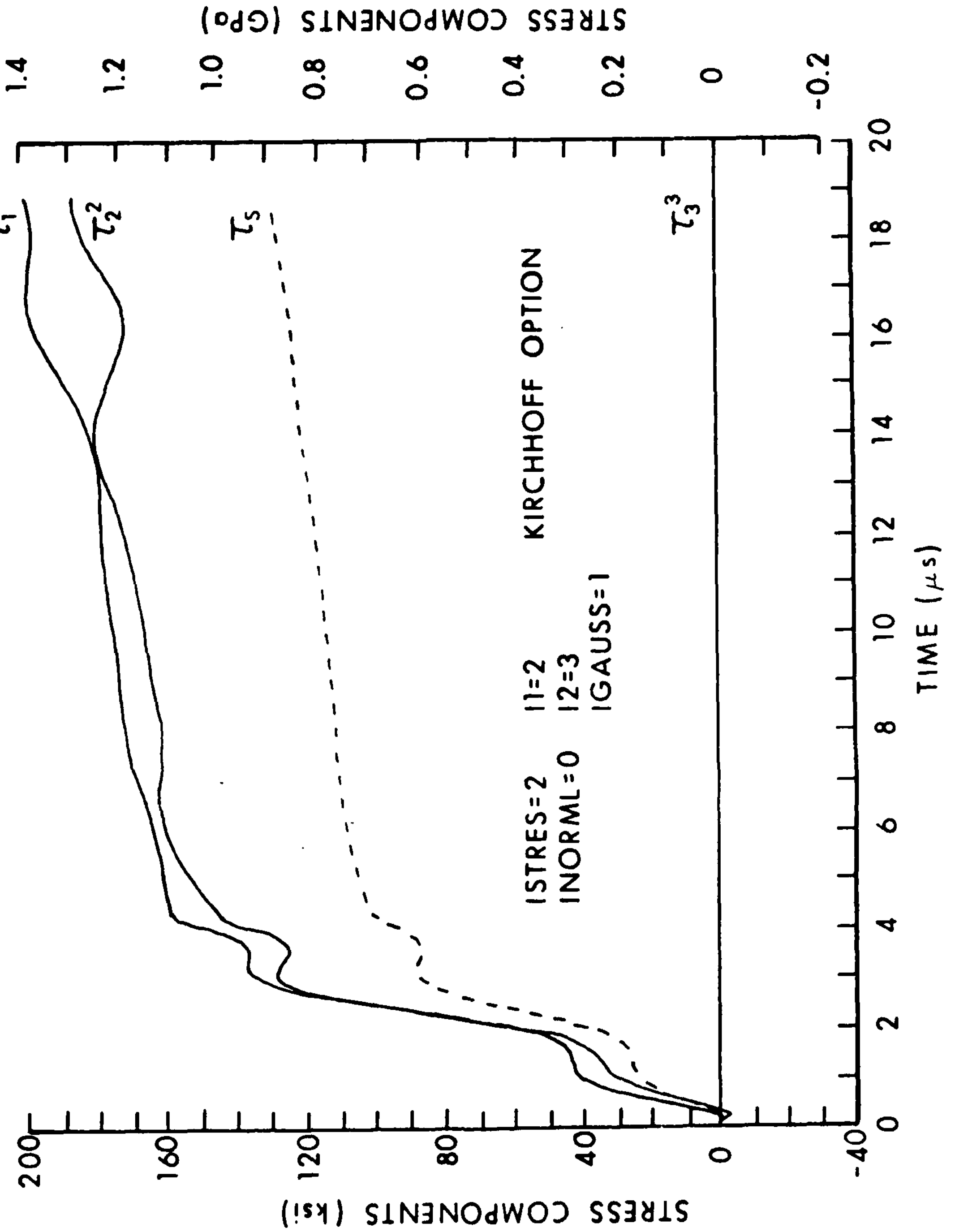


Figure 2. Transient Stress Components for Indicated Options

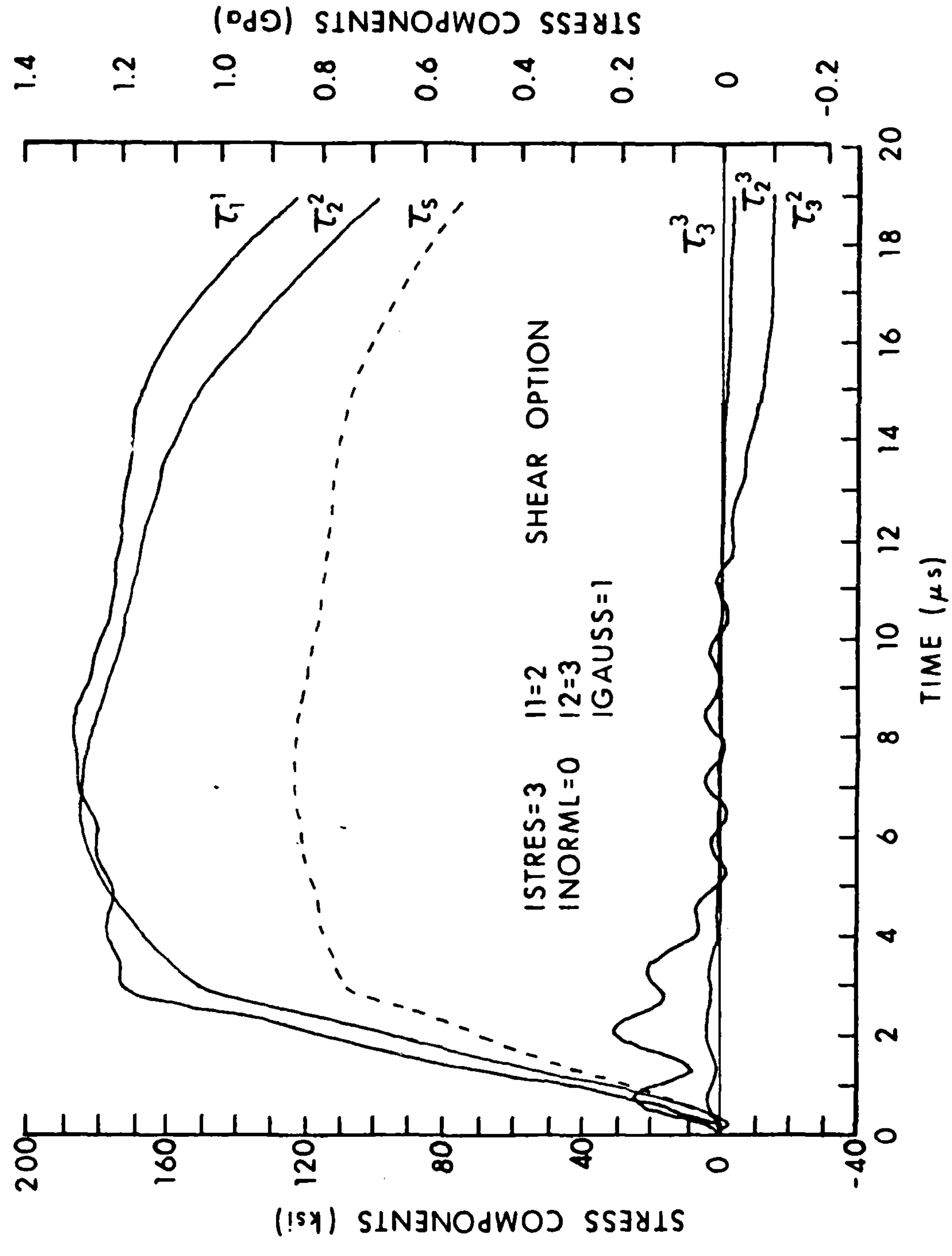


Figure 3. Transient Stress Components for Indicated Options

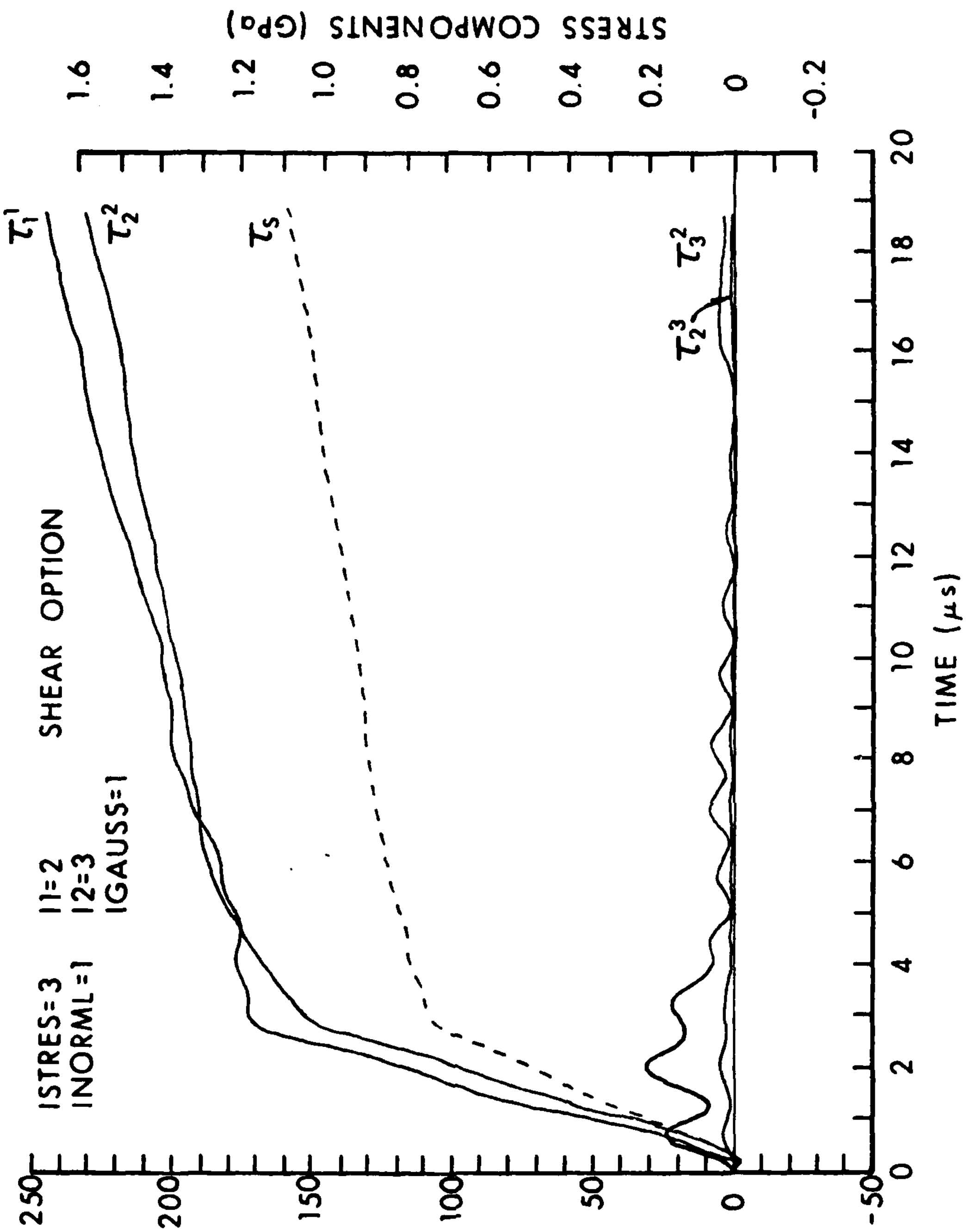


Figure 4. Transient Stress Components for Indicated Options

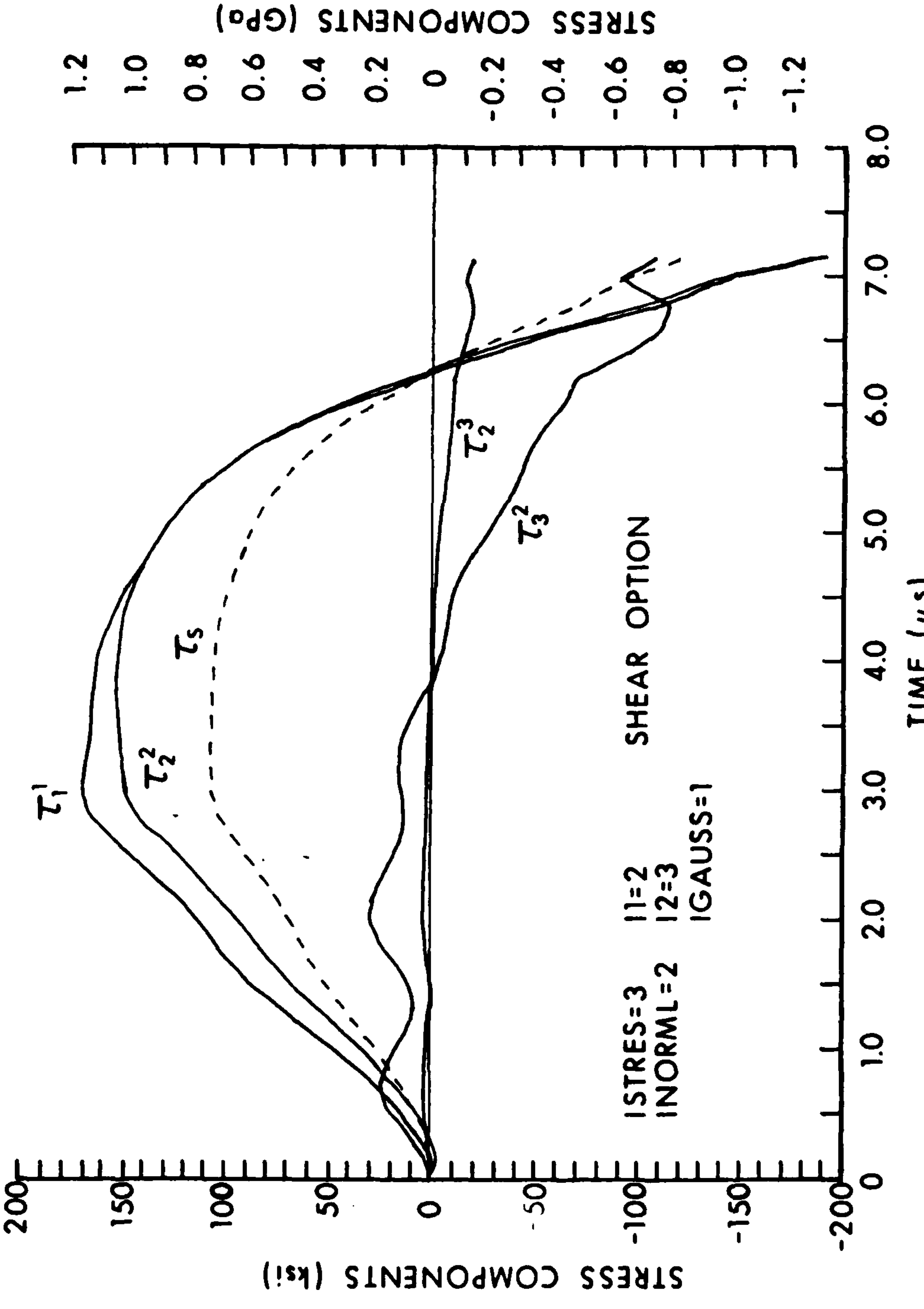


Figure 5. Transient Stress Components for Indicated Options

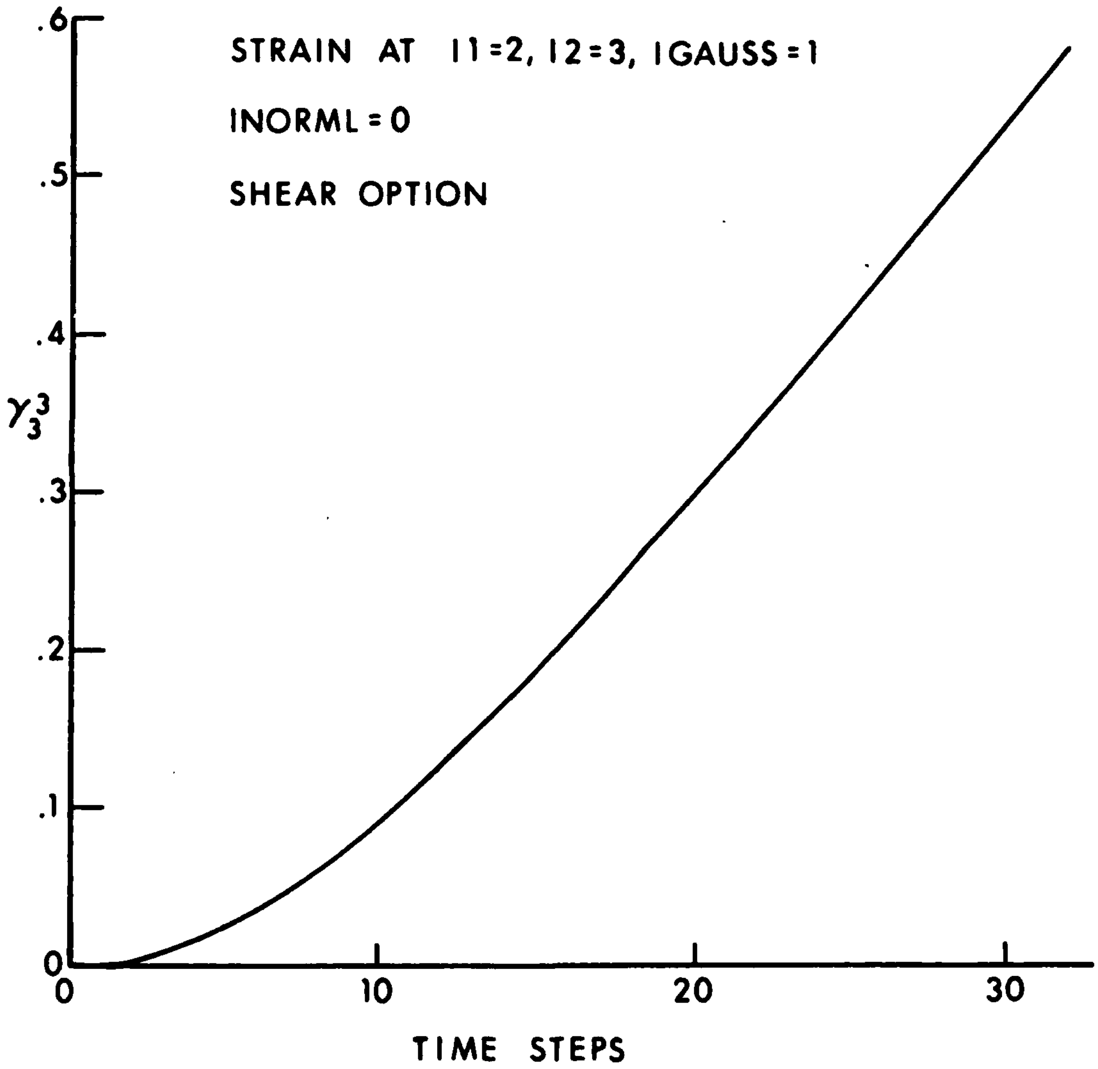


Figure 6. Growth of Through - Thickness Strain

B. Unstable Growth of D^3

As previously noted, the three independent components of the vector $\bar{D}$ represent the additional degrees-of-freedom possessed by the SHEAR option of the PETROS 4 code. With the INORML = 0 option the Cartesian components of $\bar{D}$ are determined by three equations of motion derived using a variational principle. However, it appears that this formulation provides no material stiffness-based restoring force to oppose changes in the magnitude of D^3, the component of $\bar{D}$ in the direction of the normal to the reference surface. As a consequence solutions using this option tend to exhibit a monotonic growth or decrease in the magnitude of D^3 and of the directly dependent through-thickness strain γ_3^3, see Figure 6. Corrective measures to circumvent this defect will be presented in Chapter IV.

C. Constitutive Relation for ISTRES = 3

In the course of checking the PETROS 4 code it was discovered that values of the normal components of trial stresses (TR (I, I) in the code) were being evaluated incorrectly in the STRESS subroutine for the option ISTRES = 3. Specifically, the non-zero value of DGAMMX (3, 3) was being included in the calculation of DGAMMA, which is inappropriate when the stress-strain relation is constrained by the condition $\tau_3^3 \equiv 0$. Also, non-zero values of corrector stress TC (3,3) (as well as TR (3,3) and TM (3,3)) were being used in the calculation of $\tilde{\lambda}$ and $(\tau_j^i)_{n+1}$, causing an additional error in the elasto-plastic stress evaluation. The fact that later in the cycle τ^{33} was set equal to zero did nothing to remedy the errors introduced into the other component of τ^{ij}. Once this problem was detected, appropriate corrections were readily made to the STRESS subroutine.

D. Effect of Through-Thickness Normal Stresses

For most shell structures subjected to surface loadings the magnitude of the through-thickness normal stress τ_3^3 is negligible in comparison to induced flexural and membrane stresses appearing as the components τ_1^1 and τ_2^2. However, for the presently contemplated application the blast pressure-induced values of τ_3^3 during the early portion of the loading are of the same order as the other normal stresses and deserve to be taken into account when applying the constitutive relations. This raises the question as to whether such problems can be treated in a rational manner without resorting to a complete three-dimensional analysis. It should be noted that, of the options available with the PETROS 4 code, only the ISTRES = 0 option does not set $\tau_3^3 \equiv 0$. Unfortunately, as shown in Figure 1, solutions obtained using this option predict unreasonably large tensile values of τ_3^3 rather than the compressive stresses which would be expected to result from surface pressure loading. An alternative formulation for incorporating the effects of through-thickness normal stresses will be given in Chapter III.

E. Omission of Surface Traction Effects

In the theoretical formulation report[1] for the PETROS 4 code the effect of surface tractions was embodied in the equations of motion by terms designated $\tilde{E}^k_{(n)}$. Later in the same report it was argued that the terms $\tilde{E}^k_{(2)}$ and $\tilde{E}^k_{(3)}$ could be neglected for thin shells. However in Appendix D of Reference 1, where the equations of motion for the SHEAR (moderately thick shell) equations are presented, the term $\tilde{E}^k_{(2)}$ is retained (as it should be). It was discovered that this term was not included in the finite difference equations of motion in subroutine EQUIL2 of the PETROS 4 code which are used to calculate the components of $\overline{D}$.

F. Reconstitution of Mixed Tensor Stresses

In the cyclic time marching solution procedure employed by the PETROS 4 code the values of the unsymmetric mixed tensor stress components τ^j_k in each sublayer at the previous time step are needed in the calculation of elastoplastic stresses at the current time step. However, in an apparent effort to economize on use of computer memory, the symmetric contravariant stress tensor components τ^{ij} are saved rather than the mixed tensor components. Thus at the previous time step the calculations $\tau^{ij} = G^{ik}\tau^j_k$ are performed and, when stress calculations are resumed at the next cycle, the values of τ^j_k are reconstituted by use of $\tau^j_k = G_{ki}\tau^{ij}$. However, in the interim a new set of metric tensors has been calculated so that the reconstituted values of τ^j_k are not generally identical with the values determined during the previous cycle. In fact, if a significant geometry change has occurred the differences may be appreciable. It is feared that, for long computer runs, these differences may accumulate to cause serious departures from the correct solution.

G. Plastic Work

The PETROS 4 code calculates the total plastic work performed within the boundaries of the structure as one of the ingredients of an energy balance diagram which is useful for detecting numerical instabilities and for determining an appropriate time to terminate the solution. The other ingredients are the total kinetic energy, total elastic strain energy, and the total work done on the structure by external loads. For conservation of energy the sum of the kinetic energy, strain energy, and plastic work should not exceed the external work, except possibly for a small discretization error. However, for the blast loaded panels of current interest the energy balance diagram of Figure 7 is typical. By a process of elimination it has been concluded that the observed discrepancy is due to an error in the plastic work, either in the finite deformation formulation or the coding. Fortunately, the computation of plastic work is an auxiliary calculation which has no effect on the basic solution process.

The next three chapters are devoted to modifications to the PETROS 4 code designed to remedy the foregoing deficiencies.

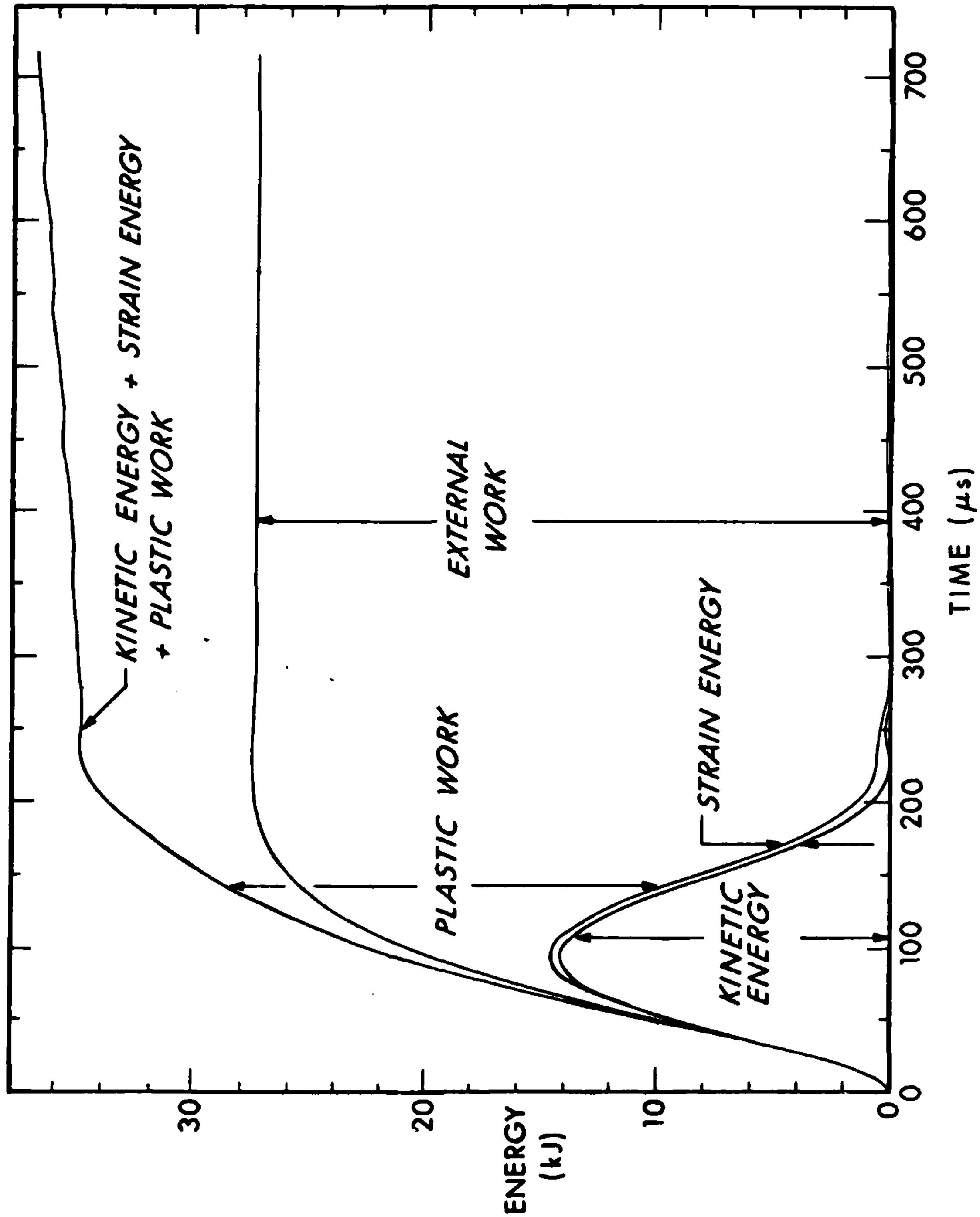


Figure 7. Energy Balance Diagram

III. PRESCRIBED VARIATION OF THROUGH-THICKNESS NORMAL STRESS

The issues raised in Section D of Chapter II concerning the effects of through-thickness normal stresses will now be examined in more detail.

A. Stress Wave Considerations

In order to assess the significance of the τ_3^3 stress component on elastoplastic calculations it is necessary to have some information as to the manner in which this component varies through the shell thickness. Therefore, consideration was given to an idealized one-dimensional problem of elastic stress wave propagation for the through-thickness direction. This is not as restrictive as it may seem. One is not concerned with blast pressures great enough to induce plasticity in the first pass of the stress wave through the thickness or to cause spallation off the far side of the shell because it is known that rupture of the shell will occur for lower blast pressures. The analysis which follows is for only slightly more than two wave passes through a plate and it is known that plasticity is not induced in the plate (due to flexure and stretching) until much later. However, it will be possible to draw conclusions which will also apply during elastoplastic response.

In the idealized problem it was assumed that the upper surface of a plate was subjected to a uniformly distributed blast pressure which jumped to a value p_o followed by an exponential decay. The lower surface of the plate was assumed stress-free. The solid line in Figure 8 is the traveling wave solution for the through-thickness normal stress at the Gauss point closest to the loaded surface. On the other hand, if one assumes a linear variation of through-thickness stress from $-p(t)$ at the upper surface to zero at the lower surface the stress at each Gauss point can be calculated. In this manner the dashed curve in Figure 8 was obtained (this curve is actually the "variable mean" of the traveling wave solution). Similar results can be obtained at the other Gauss points. The one-dimensional traveling wave analysis is only applicable for a uniform blast pressure which is not the actual distribution for an explosive charge detonated near a plate. Inasmuch as it is desired to avoid a general three-dimensional response analysis it is felt that the linear variation of through-thickness normal stress represents a reasonably satisfactory approximate basis for defining this component of the stress tensor in subsequent calculations; certainly this is better than assuming $\tau_3^3 \equiv 0$.

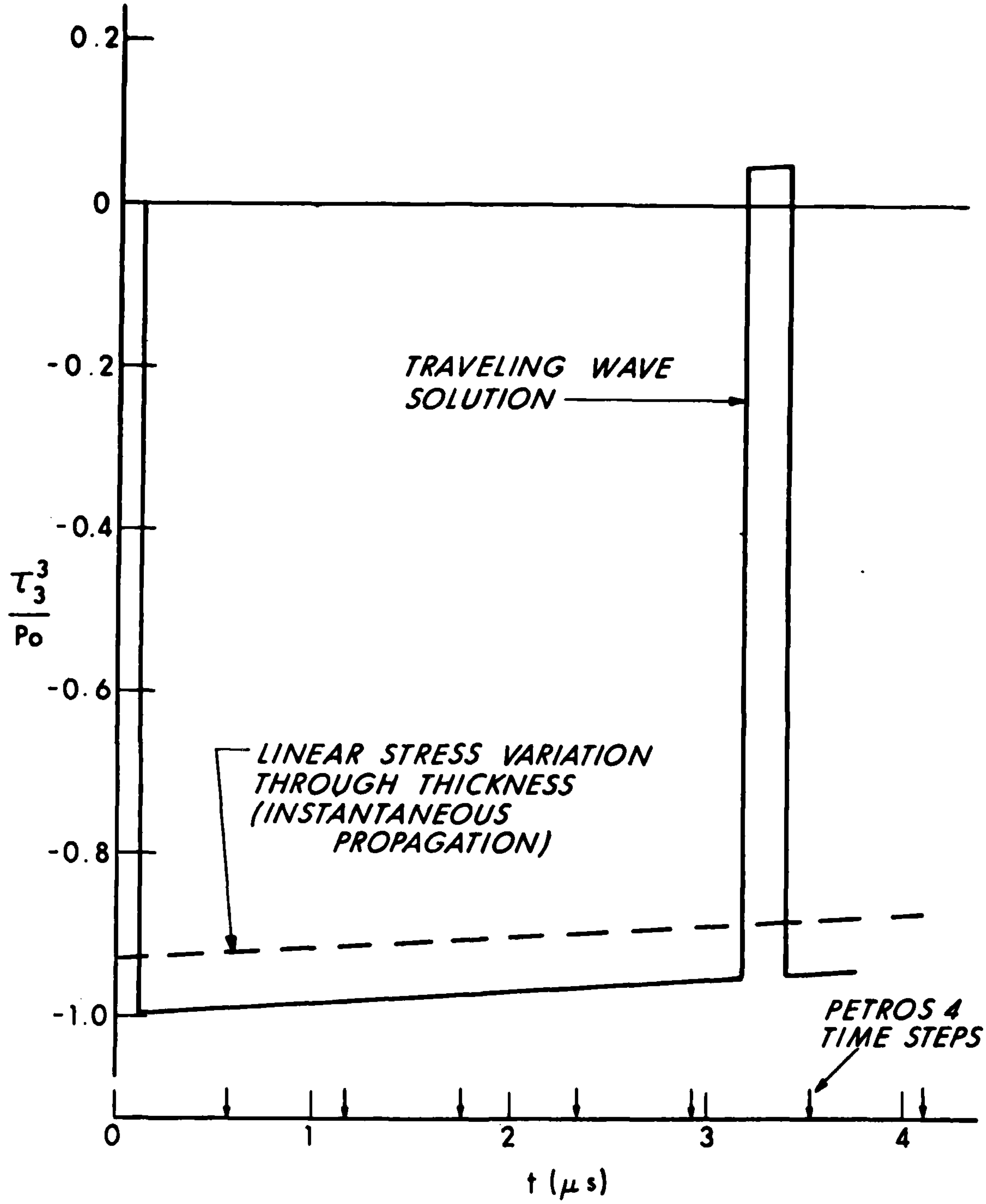


Figure 8. Stress τ_3^3 at Gauss Point 4

B. Modified Constitutive Relation

At the upper surface of the plate (or shell) the boundary conditions* are

$$\tau^{33}(\xi^\alpha, \frac{h}{2}, t) = -G^{33}(\xi^\alpha, \frac{h}{2}, t)\, p\,(\xi^\alpha, \frac{h}{2}, t) \quad , \quad \tau^{31} = \tau^{32} = 0 \tag{1}^{**}$$

while on the lower surface ($\zeta = -h/2$) all three stress components vanish. The mixed tensor through-thickness component at the upper surface is given by

$$\begin{aligned} \tau_3^3(\xi^\alpha, \frac{h}{2}, t) &= G_{3j}(\xi^\alpha, \frac{h}{2}, t)\, \tau^{j3}(\xi^\alpha, \frac{h}{2}, t) \quad \dagger \\ &= G_{33}(\xi^\alpha, \frac{h}{2}, t)\, \tau^{33}(\xi^\alpha, \frac{h}{2}, t) \\ &= -G_{33}(\xi^\alpha, \frac{h}{2}, t)\, G^{33}(\xi^\alpha, \frac{h}{2}, t)\, p(\xi^\alpha, \frac{h}{2}, t) \end{aligned} \tag{2}$$

For the desired linear variation of τ_3^3 between the surface boundary conditions,

$$\tau_3^3(\xi^\alpha, \zeta_g, t) = -G_{33}(\xi^\alpha, \frac{h}{2}, t)\, G^{33}(\xi^\alpha, \frac{h}{2}, t)\, p(\xi^\alpha, \frac{h}{2}, t)\, (\frac{1}{2} + \frac{\zeta_g}{h}) \tag{3}$$

where ζ_g are the locations of the discrete Gauss points.

In the PETROS 4 code the nine incremental stress components are determined from nine strain increments $\Delta\gamma_j^i$ and the known values of the stress components $(\tau_j^i)_n$ at the time t_n prior to the incremental change. When the stress component τ_3^3 is prescribed as shown in equation (3) its value is known at both t_n and t_{n+1} so that

$$\Delta\tau_3^3 = \left(\tau_3^3\right)_{n+1} - \left(\tau_3^3\right)_n \tag{4}$$

is also prescribed. Thus the elastoplastic constitutive problem is shifted to determining eight incremental stress components and an incremental strain component $\Delta\gamma_3^3$ which generally differs from that provided by the ZETA subroutine.

---

* *These are the physical boundary conditions. The displacement model embodied in the PETROS 4 code does not provide for satisfaction of these conditions.*

** *Superscripts and subscripts range over the integers as follows: greek 1,2; latin 1,2,3.*

† *The summation convention is employed: terms having a repeated index, once as a subscript and once as a superscript, are to be summed over the range of that index.*

The generalized Hooke's law can be rearranged to obtain the following expressions for the trial stress increments:

$$
\begin{aligned}
\overset{T}{\Delta\tau}{}_1^1 &= \frac{E}{1-\nu^2}(\Delta\gamma_1^1+\nu\Delta\gamma_2^2) + \frac{\nu}{1-\nu}\Delta\tau_3^3 \\
\overset{T}{\Delta\tau}{}_2^2 &= \frac{E}{1-\nu^2}(\nu\Delta\gamma_1^1 + \Delta\gamma_2^2) + \frac{\nu}{1-\nu}\Delta\tau_3^3 \\
\overset{T}{\Delta\tau}{}_j^i &= \frac{E}{1+\nu}\Delta\gamma_j^i \qquad \text{for } i\neq j
\end{aligned}
\tag{5}
$$

The trial stresses at time t_{n+1} are

$$\left(\overset{T}{\tau}{}_j^i\right)_{n+1} = \left(\tau_j^i\right)_n + \overset{T}{\Delta\tau}{}_j^i \tag{6}$$

To determine whether plasticity occurs during the interval $t_{n+1} - t_n = \Delta t$ the values of the trial stresses are substituted in the plastic potential function

$$\overset{T}{\phi}_{n+1} = \left[\overset{T}{\tau}{}_j^i\overset{T}{\tau}{}_i^j - \frac{1}{3}\left(\overset{T}{\tau}{}_m^m\right)^2 - \frac{2}{3}\sigma_Y^2\right]_{n+1} \tag{7}$$

which may be recognized as the von Mises yield function. If the resulting value of $\overset{T}{\phi}_{n+1} \leqslant 0$ the stress change is elastic and the trial stresses $(\overset{T}{\tau}{}_j^i)^{n+1}$ become the actual stresses $\left(\tau_j^i\right)_{n+1}$. For $\overset{T}{\phi}_{n+1} > 0$ plasticity occurs and the total strain increments are composed of

$$\Delta\gamma_j^i = \overset{e}{\Delta\gamma}{}_j^i + \overset{p}{\Delta\gamma}{}_j^i \tag{8}$$

i.e., elastic and plastic parts. The plastic strain increments are obtained from the flow rule

$$\overset{p}{\Delta\gamma}{}_j^i = \frac{\partial\phi_n}{\partial\tau_i^j}\Delta\lambda = \left[2\left(\tau_j^i\right)_n - \frac{2}{3}\left(\tau_m^m\right)_n\delta_j^i\right]\Delta\lambda \tag{9}$$

where $\Delta\lambda$ is a scalar multiplier. Even in the presence of plasticity the stress increments are related to the elastic strain increments by Hooke's law:

$$\Delta\tau_j^i = \frac{E}{1+\nu}\overset{e}{\Delta\gamma}{}_j^i + \left\{\frac{\nu E}{1-\nu^2}\overset{e}{\Delta\gamma}{}_\alpha^\alpha + \frac{\nu}{1-\nu}\Delta\tau_3^3\right\}\delta_j^i \tag{10}$$

Then, substituting

$$\overset{e}{\Delta\gamma}{}_j^i = \Delta\gamma_j^i - \overset{p}{\Delta\gamma}{}_j^i \tag{11}$$

and the plastic strain increments from equation (9), one obtains

$$\Delta\tau_j^i = \overset{T}{\Delta\tau}{}_j^i - \lambda\left(\overset{C}{\tau}{}_j^i\right)_n \tag{12}$$

where $\overset{T}{\Delta\tau}{}^i_j$ is given by equations (5),

$$\left(\overset{C}{\tau}{}^i_j\right)_n = \left(\tau^i_j\right)_n - \frac{1}{1-\nu}\left\{\frac{1-2\nu}{3}\left(\tau^m_m\right)_n + \nu\left(\tau^3_3\right)_n\right\}\delta^i_j \tag{13}$$

and $\lambda = \frac{2E\Delta\lambda}{1+\nu}$. Since $\Delta\tau^3_3$ is prescribed it is arbitrary whether $\overset{T}{\Delta\tau}{}^3_3$ or $(\overset{C}{\Delta\tau}{}^3_3)_n$ is defined as long as equation (12) is satisfied. However, it is preferable to set $\overset{T}{\Delta\tau}{}^3_3 = \Delta\tau^3_3, (\overset{C}{\Delta\tau}{}^3_3)_n = 0$ so that when the test for plasticity (equation (7)) is applied, $\left(\overset{T}{\tau}{}^3_3\right)_{n+1}$ is already known. The stresses at the end of the time interval are

$$\left(\tau^i_j\right)_{n+1} = \left(\tau^i_j\right)_n + \Delta\tau^i_j = \left(\tau^i_j\right)_n + \overset{T}{\Delta\tau}{}^i_j - \lambda\left(\overset{C}{\tau}{}^i_j\right)_n = \left(\overset{T}{\tau}{}^i_j\right)_{n+1} - \lambda\left(\overset{C}{\tau}{}^i_j\right)_n \tag{14}$$

In the mechanical sublayer constitutive model each hypothetical sublayer is treated as an elastic, perfectly plastic material having a distinct yield stress. Consequently the condition $\phi_{n+1} = 0$ is imposed in order to determine the parameter λ for each sublayer experiencing plasticity. When the stresses given by equation (14) are subjected to this condition a quadratic equation of the form

$$A\lambda^2 - 2B\lambda + C = 0 \tag{15}$$

results, where

$$\begin{aligned}
A &= \left(\overset{C}{\tau}{}^i_j\right)_n\left(\overset{C}{\tau}{}^j_i\right)_n - \frac{1}{3}\left(\overset{C}{\tau}{}^m_m\right)^2_n \\
B &= \left(\overset{T}{\tau}{}^i_j\right)_{n+1}\left(\overset{C}{\tau}{}^j_i\right)_n - \frac{1}{3}\left(\overset{T}{\tau}{}^m_m\right)_{n+1}\left(\overset{C}{\tau}{}^r_r\right)_n \\
C &= \overset{T}{\phi}_{n+1} = \left(\overset{T}{\tau}{}^i_j\right)_{n+1}\left(\overset{T}{\tau}{}^j_i\right)_{n+1} - \frac{1}{3}\left(\overset{T}{\tau}{}^m_m\right)^2_{n+1} - \frac{2}{3}\sigma_y{}^2
\end{aligned} \tag{16}$$

From equation (15),

$$\lambda = \frac{B}{A} - \left\{\left(\frac{B}{A}\right)^2 - \frac{C}{A}\right\}^{1/2} \tag{17}$$

Once a real, positive value of λ has been obtained the stresses $(\tau^i_j)_{n+1}$ can be calculated by use of equation (14). Huffington[6] has presented a procedure for dealing with complex

---

[6] *N. J. Huffington, Jr., "Numerical Analysis of Elastoplastic Stresses," US Army Ballistic Research Laboratory Memorandum Report No. 2006, September 1969.* AD 861688.

roots should they arise by subdividing Δt for purposes of stress evaluation only. This procedure was already incorporated in the PETROS 4 code. However, it has been found that under certain circumstances even when a real root λ is obtained without subdividing Δt, inaccurate or oscillatory stresses may result. This difficulty is associated with an excessively large excursion of the trial stress vector $\overset{T}{\tau}$ outside the yield surface in stress space. A technique for coping with this problem devised by Huffington (see Appendix B of Reference 7) has recently been introduced into the PETROS 4 code. It entails calculating an integer L which defines the number of subdivisions of the time step Δt, where

$$L = \mathrm{INT}\left[\mathrm{YLDFAC}\left\{\sqrt{(1.5\overset{T}{\phi}_{n+1} + \sigma_y^2)/\sigma_y^2} - 1\right\}\right] + 1 \tag{18}$$

YLDFAC is a parameter which varies the accuracy of the stress evaluation; it ranges from 0 (no subdivision of Δt) to ∞ (differential subintervals). Normally YLDFAC = 1 is used.

The foregoing formulation, entailing the prescribed linear variation of τ_3^3 through the thickness, has been incorporated in the PETROS 4 code as option ISTRES = 4.

C. Through-Thickness Strain Calculation

The strain increment component $\Delta\gamma_3^3$ may be determined by use of

$$\Delta\gamma_3^3 = \frac{1}{E}\left[\Delta\tau_3^3 - \nu\left(\Delta\tau_1^1 + \Delta\tau_2^2\right)\right] + \frac{(1+\nu)}{3E}\left(2\tau_3^3 - \tau_1^1 - \tau_2^2\right)_n\lambda \tag{19}$$

once λ has been evaluated. If plasticity is occurring at a Gauss point the value of $\Delta\gamma_3^3$ will generally be different for each sublayer. A weighted average $\Delta\gamma_3^3$ for the Gauss point can be obtained through multiplying the sublayer $\Delta\gamma_3^3$ strains by the same weighting factors employed with the mechanical sublayer model and summing. Alternatively, after the total elastoplastic stresses τ_j^i at a Gauss point have been determined the value of $\Delta\gamma_3^3$ for the Gauss point can be calculated by imposing the condition of plastic incompressibility:

$$\Delta\gamma_3^3 = \frac{\Delta\tau_k^k}{3K} - \Delta\gamma_\alpha^\alpha \tag{20}$$

Generally, a different value of $\Delta\gamma_3^3$ will result for each Gauss point rather than the constant value of $\Delta\gamma_3^3$ through the thickness which the PETROS 4 displacement function admits. In the next chapter a compromise resolution of this discrepancy will be presented.

---

[7] *J. M. Santiago, H. L. Wisniewski, and N. J. Huffington, Jr., "A User's Manual for the REPSIL Code," US Army Ballistic Research Laboratory Report No. 1744, October 1974.* AD A003176.

IV. MODIFICATION OF THE D^3 DEGREE-OF-FREEDOM

The problem of unstable growth of the variable $D^3(\xi^\alpha,t)$ when the INORML = 0 option is employed was discussed in Chapter II. This variable corresponds to just one of the six degrees-of-freedom which appear in the PETROS 4 displacement function, which has the following forms:

$$\bar{u}(\xi^\alpha,\zeta,t) = \bar{u}_o(\xi^\alpha,t) + \zeta[\bar{N}(\xi^\alpha,t) - \bar{n}(\xi^\alpha,0)] + \zeta\bar{D}(\xi^\alpha,t) \tag{21a}$$

$$= v^\alpha\bar{A}_\alpha + w\bar{N} + \zeta(\bar{N} - \bar{n}) + \zeta(D^\alpha\bar{A}_\alpha + D^3\bar{N}) \tag{21b}$$

$$= Y^{(1)k}\bar{i}_k + \zeta N^k\bar{i}_k + \zeta Y^{(2)k}\bar{i}_k - (\bar{r}_o + \zeta\bar{n}) \tag{21c}$$

The first form is a vectorial one, showing the independent variables upon which these vectors depend. Equation (21b) designates the components of the vector quantities in the directions of the basis vectors $\bar{A}_\alpha$ lying in the deformed reference surface and the normal vector $\bar{N}$ perpendicular to this surface. The third form represents the rectangular cartesian component version of the same displacement function. Note that

$$\bar{D}(\xi^\alpha,t) = D^\alpha\bar{A}_\alpha + D^3\bar{N} = Y^{(2)k}\bar{i}_k \tag{22}$$

When the strain-displacement relations are applied to the displacement model of equation (21) it is found that $\Delta\gamma_{33}$ and γ_{33} cannot vary with the through-thickness coordinate ζ. This result is of course in direct contradiction to the conclusion regarding the variability of strain components through the thickness reached at the end of the preceding chapter, where these quantities were evaluated using the constitutive relations. Clearly, a generalization of the displacement function to permit modeling the variation of γ_3^3 with ζ would be desirable. However, this would entail an extensive reformulation for the PETROS 4 code with the addition of other degrees-of-freedom and increased storage and computing requirements. Since this is not feasible at present it appears that the best one can do is to modify the value of $D^3(\xi^\alpha,t)$ so that the value of $\Delta\gamma_3^3$ obtained by differentiating the displacement function will agree with some mean value of the $\Delta\gamma_3^3$ strain increments at the ξ^α location which are obtained using the constitutive relations (i.e., equations (19) or (20)).

A. Modifications to Cartesian Components

Consider that the quantities $Y^{(2)j}$ are known at a time t_n and that the EQUIL2 subroutine of PETROS 4 has produced the next set of incremental changes $\Delta Y^{(2)j}$. These incremental changes also satisfy an equation similar to equation (22):

$$\overline{\Delta D} = \Delta D^\alpha\bar{A}_\alpha + \Delta D^3\bar{N} = \Delta Y^{(2)j}\,\bar{i}_j \tag{23}$$

After modifications to certain incremental quantities the following equation will apply:

$$\overline{\overset{m}{\Delta D}} = \Delta D^\alpha\,\bar{A}_\alpha + \overset{m}{\Delta D^3}\,\bar{N} = \overset{m}{\Delta Y^{(2)j}}\,\bar{i}_j \tag{24}$$

where the terms with overscript m are modified. Note that the terms involving $\overline{A}_\alpha$ are unmodified, also that $\overline{N}$ depends only on $Y^{(1)j}$ and is unaffected by changes in $\overline{D}$. Taking the inner product of both sides of equations (23) and (24) with the unit vector $\vec{i}^j$ one obtains:

$$\Delta Y^{(2)j} = \Delta D^\alpha \overline{A}_\alpha \cdot \vec{i}^j + \Delta D^3 N^j \tag{25}$$

$$\overset{m}{\Delta Y}{}^{(2)j} = \Delta D^\alpha \overline{A}_\alpha \cdot \vec{i}^j + \overset{m}{\Delta D}{}^3 N^j \tag{26}$$

Letting

$$\eta(\xi^\alpha, t) = \overset{m}{\Delta D}{}^3 - \Delta D^3 \tag{27}$$

and subtracting equation (25) from equation (26) gives

$$\overset{m}{\Delta Y}{}^{(2)j} = \Delta Y^{(2)j} + \eta N^j \tag{28}$$

The modified values of the cartesian components of $\overline{D}$ at the next time step are

$$\begin{aligned}(\overset{m}{Y}{}^{(2)j})_{n+1} &= (Y^{(2)j})_n + \overset{m}{\Delta Y}{}^{(2)j} \\ &= (Y^{(2)j})_n + \Delta Y^{(2)j} + \eta N^j \\ &= (Y^{(2)j})_{n+1} + \eta N^j\end{aligned} \tag{29}$$

Before proceeding further a formulation for η which will produce the desired effect is needed.

B. Strain Equivalence Criterion

The derivation of an expression for determining η requires examination of the non-linear incremental strain-displacement relations employed by PETROS 4. The incremental through-thickness strain $\Delta\gamma_3^3$ is related to the covariant incremental strains by

$$\Delta\gamma_3^3 = \Delta\gamma_{3m} G^{m3} \tag{30}$$

In turn, the covariant strain increment $\Delta\gamma_{33}$ is related to the cartesian components of the basis vector $\overline{G}_3$ and its incremental change by

$$\Delta\gamma_{33} = J_3^j \Delta J_3^j - 0.5\, \Delta J_3^j \Delta J_3^j \tag{31*}$$

---

* *Since these are cartesian components the summation convention applies even though the repeated indices are superscripts.*

For the SHEAR option (only),

$$J_3^j = N^j + Y^{(2)j} \tag{32}$$

$$\Delta J_3^j = \Delta N^j + \Delta Y^{(2)j} \tag{33}$$

Using these relations the effect of the previously cited modifications can readily be traced. As noted before, the surface normal depends only on the reference surface $Y^{(1)j}$ so that modifications to $\overline{D}$ have no effect on N^j or ΔN^j.

When $\Delta Y^{(2)j}$ is modified as indicated by equation (28) the components of the basis vector increment $\overline{\Delta G_3}$ are affected as follows:

$$\begin{aligned} \overset{m}{\Delta J_3^j} &= \Delta N^j + \Delta Y^{(2)j} + \eta N^j \\ &= \Delta J_3^j + \eta N^j \end{aligned} \tag{34}$$

Similarly,

$$\overset{m}{J_3^j} = J_3^j + \eta N^j \tag{35}$$

The effect on $\Delta\gamma_{33}$ is, by use of equations (31), (34), and (35),

$$\begin{aligned} \overset{m}{\Delta\gamma_{33}} &= \overset{m}{J_3^j}\,\overset{m}{\Delta J_3^j} - 0.5\,\overset{m}{\Delta J_3^j}\,\overset{m}{\Delta J_3^j} \\ &= (J_3^j + \eta N^j)(\Delta J_3^j + \eta N^j) - 0.5(\Delta J_3^j + \eta N^j)^2 \\ &= \Delta\gamma_{33} + J_3^j N^j \eta + 0.5\eta^2 \end{aligned} \tag{36}$$

Solving this quadratic expression for η ,

$$\eta = -J_3^j N^j + \left[\left(J_3^j N^j\right)^2 - 2\left(\Delta\gamma_{33} - \overset{m}{\Delta\gamma_{33}}\right)\right]^{1/2} \tag{37}$$

From equation (30) it follows that

$$\Delta\gamma_{33} = \frac{\Delta\gamma_3^3 - \Delta\gamma_{31}G^{13} - \Delta\gamma_{32}G^{23}}{G^{33}} \tag{38}$$

and a similar form for $\overset{m}{\Delta\gamma_{33}}$. Substituting these into equation (37):

$$\eta = -J_3^j N^j + \left[\left(J_3^j N^j\right)^2 - 2\left(\frac{\Delta\gamma_3^3 - \Delta\gamma_{31}G^{13} - \Delta\gamma_{32}G^{23}}{G^{33}} - \frac{\overset{m}{\Delta\gamma_3^3} - \overset{m}{\Delta\gamma_{31}}\overset{m}{G^{13}} - \overset{m}{\Delta\gamma_{32}}\overset{m}{G^{23}}}{\overset{m}{G^{33}}}\right)\right]^{1/2} \tag{39}$$

Up to this point no approximations have been made; however, since the quotient involving modified quantities contains several unknowns it is useful to assume

$$\overset{m}{G}{}^{33} \approx G^{33}\ ,\ \overset{m}{\Delta}\gamma_{31}\overset{m}{G}{}^{13} + \overset{m}{\Delta}\gamma_{32}\overset{m}{G}{}^{23} \approx \Delta\gamma_{31}\, G^{13} + \Delta\gamma_{32}\, G^{23}$$

On this basis equation (39) reduces to

$$\eta = -J_3^j N^j + \left[\left(J_3^j N^j\right)^2 - 2\left(\frac{\Delta\gamma_3^3 - \overset{m}{\Delta}\gamma_3^3}{G^{33}}\right)\right]^{1/2} \tag{40}$$

which will provide the proper η to produce the desired value of $\overset{m}{\Delta}\gamma_3^3$.

C. Implementation of the Modification to D^3

The normal sequence of calculations in the PETROS 4 code is indicated in the simplified flow chart of Figure 9 by solid lines. Beginning at subroutine EQUIL2, where values of $\Delta Y^{(2)j}$ and $(Y^{(2)j})_{n+1}$ are determined, the calculations proceed through subroutines GEOMET, STRAIN, and ZETA during which the unmodified geometric quantities N^j, ΔN^j, J_k^j, ΔJ_k^j, G_{jk}, G^{jk}, $\Delta\gamma_{jk}$ and $\Delta\gamma_k^j$ * are computed. In subroutine ZETA there is a call for subroutine STRESS where the elastoplastic stress calculations are made, after which there would normally be a return to subroutine ZETA for calculation of force and moment resultants at t_{n+1}. However, to implement the desired modification to D^3 the following changes have been incorporated into PETROS 4.

Once the elastoplastic stress increments and stresses $(\tau_k^j)_{n+1}$ have been determined in subroutine STRESS, equation (19) is used to calculate the strain increment $\Delta\gamma_3^3$ for each sublayer. The Gauss point values of $\Delta\gamma_3^3$ are then determined using the mechanical sublayer coefficients. After returning to subroutine ZETA a Gaussian mean value of $\Delta\gamma_3^3$ through the thickness is computed which is then introduced into equation (40) as $\overset{m}{\Delta}\gamma_3^3$, thus completing the information necessary to evaluate η . The program then branches back to a point near the end of subroutine EQUIL2 (see dotted path on Figure 9) where $\overset{m}{\Delta}Y^{(2)j}$ and $(\overset{m}{Y}{}^{(2)j})_{n+1}$ are calculated using equations (28) and (29). The program then proceeds forward, calculating modified values of J_k^j, ΔJ_k^j, G_{jk}, G^{jk}, $\Delta\gamma_{jk}$, and $\Delta\gamma_k^j$. Clearly, the foregoing procedure could be continued iteratively to cause the difference $\Delta\gamma_3^3 - \overset{m}{\Delta}\gamma_3^3$

---

* *While $\Delta\gamma_{33}$ is independent of ζ the use of equation (30) introduces a very slight variation of $\Delta\gamma_3^3$ with ζ. To avoid ambiguity a through-thickness Gaussian average value of $\Delta\gamma_3^3$ is calculated for use in equation (40). A similarly averaged value of G^{33} is also calculated for this purpose.*

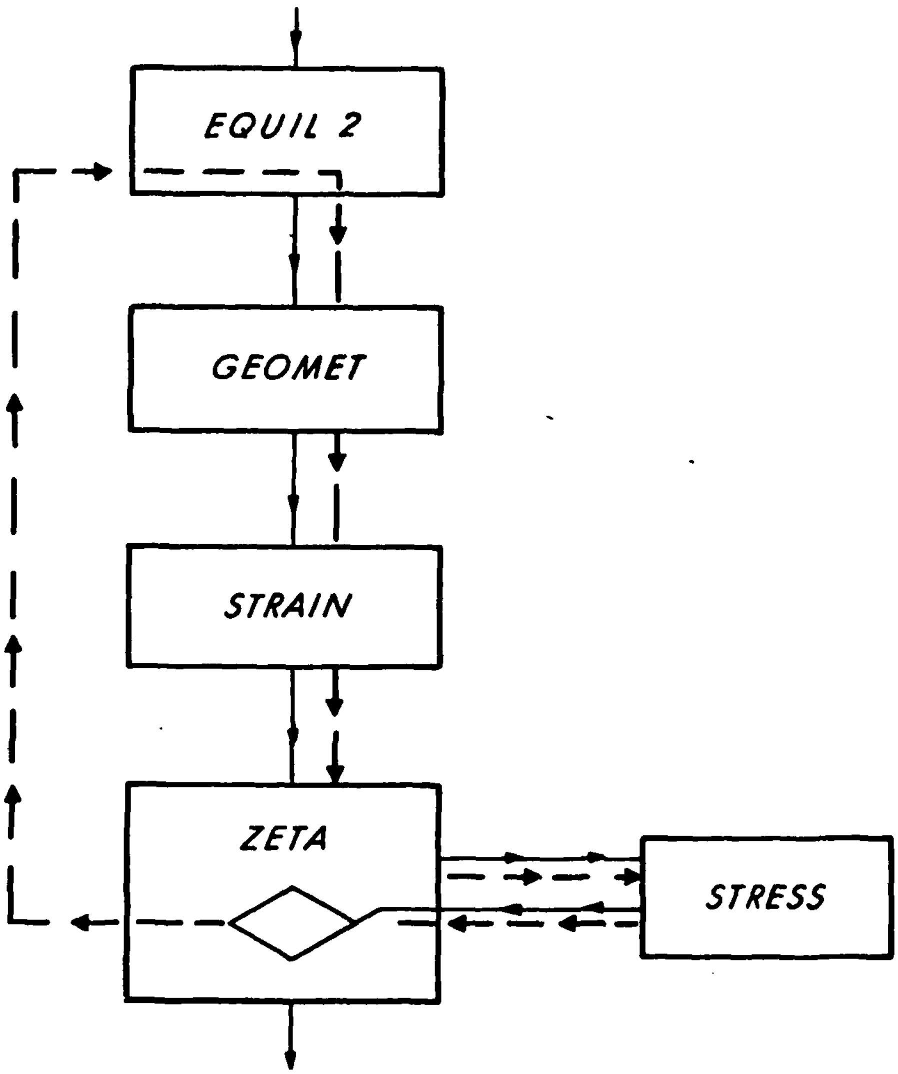


Figure 9. Simplified Flow Chart Showing Recycling Option

to be less than or equal to some small quantity. However, experience has shown that a single recycling back to EQUIL2 for the cited modification is sufficient to control the growth of D^3 and provide the desired compromise value of γ_3^3. This single recycling step per time step has been incorporated in PETROS 4 as option INORML = 3.

V. ADDITIONAL MISCELLANEOUS PROGRAM CHANGES

A. Addition of Surface Traction Terms

The omission of surface traction terms in the PETROS 4 code was discussed in Section E of Chapter II. These terms were introduced in a general manner by equation (2.71b) of the theoretical formulation report.[1] The specific relations needed to compute values of these quantities are:*

$$\tilde{E}^k_{(1)} = -\left[\sqrt{G}\, G^{3i}\, J_i^k\, p\right]_{-h/2}^{h/2} \tag{41}$$

$$\tilde{E}^k_{(2)} = \tilde{E}^k_{(3)} = -\left[\sqrt{G}\, G^{3i}\, J_i^k\, \zeta\, p\right]_{-h/2}^{h/2} \tag{42}$$

The evaluation of these expressions has been introduced into the PETROS 4 code through subroutine SFORCE. The finite difference equations of motion in subroutines EQUIL and EQUIL2 were modified to incorporate the values of $\tilde{E}^k_{(1)}$ and $\tilde{E}^k_{(2)}$, respectively.

It was hoped that the addition of the surface traction terms would eliminate the need for the one-step recycling option. Such was not found to be true and both modifications are required for a satisfactory solution, at least when the ISTRES = 4, INORML = 3 option combination is employed.

B. Symmetry of the Contravariant Stress Components

From equilibrium considerations the contravariant stress tensor τ^{ij} must be symmetric (in the absence of couple stresses) which permits the storing of six rather than nine quantities at all locations where values of this tensor must be saved for use at the next time step. However, the calculations in the STRESS subroutine which provide values of τ^{ij} do not satisfy this requirement exactly. This was true even before the introduction of the ISTRES = 4 option, for which the problem is aggravated since the prescribed value of τ_3^3 generally differs from that which would be calculated using the symmetric $\Delta\gamma_{kj}$ strain increment tensor. After the elastoplastic stresses τ_k^j have been determined and the relation $\tau^{ij} = G^{ik}\, \tau_k^j$ employed the resulting stresses τ^{ij} are generally not equal to τ^{ji} for $i \neq j$.

---

* *I am indebted to my colleague, Dr. J. M. Santiago, Jr., for providing the formulation of these expressions.*

In the original version of PETROS 4 this problem was dealt with by selecting τ^{12}, τ^{13}, and τ^{23} as the correct off-diagonal terms and equating τ^{21}, τ^{31}, and τ^{32} to these quantities, respectively. It was felt that this procedure could bias the problem solution so the program was modified to calculate all nine components τ^{ij} and then average the respective symmetrically off-diagonal components; i.e.,

$$\bar{\tau}^{ij} = \bar{\tau}^{ji} = (\tau^{ij} + \tau^{ji})/2 \qquad \text{for} \quad i \neq j \tag{43}$$

This modification was found to have a slight but not entirely negligible effect upon lengthy solutions.

C. Storage of Mixed Tensor Stresses

The problems associated with the reconstitution of the τ_k^j sublayer stresses each time step as is done in the original version of PETROS 4 were discussed in Section F of Chapter II. A revised version of this code has been developed in which the τ_k^j stresses are saved for use at the next time step rather than the τ^{jk} stresses. This version requires a 14% increase in computer memory but features a 14% reduction in running time for a representative length run. The revised version is preferred because (a) computer memory is not critical today and any reduction in running time is appreciated, (b) the cumulative deviations from the true solution associated with reconstituting the τ_k^j are circumvented, and (c) this version provides flexibility for future applications involving material failure. The required changes to the code are primarily confined to subroutine STRESS, which is listed in the Appendix.

D. Additions to Printed Output

The format for listing input data on cards for the original version of the PETROS 4 code is presented on pp. 110-129 of the user's manual[2]. In sequel, information is provided concerning modifications or additions to input data controlling various options for printed output.

A useful feature which has been added is a listing at the end of a run of the maximum and minimum values of each mixed tensor stress component along with the locations and times at which these extreme values occur. The format for this output is illustrated in Figure 10. For some applications this may provide all the desired information but, if not, it directs attention to critical locations where a re-run can provide detailed printed and plotted output. Card 5 enters the values of fifteen variables in format (15I5). The first of these variables is MAUXIL, which controls the printing of the max/min values: 0 = no print, 1 = print.

```
MAXIMUM MIXED TENSOR STRESSES

TIME=  .197988E-03 ITIME=  337 I1= 2 I2= 5 GAUSS PT. 2 TAUF(1,1)=  .20002618E+06
TIME=  .233531E-02 ITIME= 3975 I1=20 I2=16 GAUSS PT. 2 TAUF(1,2)=  .30704287E+06
TIME=  .192524E-02 ITIME= 3277 I1=20 I2=12 GAUSS PT. 4 TAUF(1,3)=  .93637187E+05
TIME=  .233531E-02 ITIME= 3975 I1=16 I2=20 GAUSS PT. 2 TAUF(2,1)=  .30704287E+06
TIME=  .197988E-03 ITIME=  337 I1= 5 I2= 2 GAUSS PT. 2 TAUF(2,2)=  .20002618E+06
TIME=  .192524E-02 ITIME= 3277 I1=12 I2=20 GAUSS PT. 4 TAUF(2,3)=  .93637187E+05
TIME=  .169788E-03 ITIME=  289 I1= 7 I2= 5 GAUSS PT. 4 TAUF(3,1)=  .32815313E+05
TIME=  .169788E-03 ITIME=  289 I1= 5 I2= 7 GAUSS PT. 4 TAUF(3,2)=  .32815313E+05
TIME=  .164324E-02 ITIME= 2797 I1= 3 I2= 3 GAUSS PT. 4 TAUF(3,3)=  .16607912E+02

MINIMUM MIXED TENSOR STRESSES

TIME=  .229125E-04 ITIME=   39 I1= 2 I2= 2 GAUSS PT. 4 TAUF(1,1)=-.28573178E+06
TIME=  .220606E-02 ITIME= 3755 I1=20 I2=16 GAUSS PT. 1 TAUF(1,2)=-.27961387E+06
TIME=  .159683E-02 ITIME= 2718 I1=20 I2= 2 GAUSS PT. 1 TAUF(1,3)=-.13696105E+06
TIME=  .220606E-02 ITIME= 3755 I1=16 I2=20 GAUSS PT. 1 TAUF(2,1)=-.27961387E+06
TIME=  .229125E-04 ITIME=   39 I1= 2 I2= 2 GAUSS PT. 4 TAUF(2,2)=-.28573178E+06
TIME=  .159683E-02 ITIME= 2718 I1= 2 I2=20 GAUSS PT. 1 TAUF(2,3)=-.13696105E+06
TIME=  .159624E-02 ITIME= 2717 I1=20 I2= 2 GAUSS PT. 1 TAUF(3,1)=-.34152247E+05
TIME=  .159624E-02 ITIME= 2717 I1= 2 I2=20 GAUSS PT. 1 TAUF(3,2)=-.34152247E+05
TIME=  .152750E-04 ITIME=   26 I1= 2 I2= 2 GAUSS PT. 4 TAUF(3,3)=-.21151544E+06
```

Figure 10. Sample of Max/Min Stress Output

The PETROS 4 code controls the printing of groups of output data through the entries on card 6a, which contains the variables

KF,IOUT(K) (K=1,KF) Format (16I5)

KF = number of print options available (currently = 14) and IOUT(K) is the cyclic frequency at which the K^{th} print option is to be printed. To avoid the printing of the K^{th} option, set IOUT(K) to an integer greater than the final time step = ITIMEF. The first eleven print options have not been changed.

Print option K=12 has been modified to provide a rather extensive output of geometric and stress variables which is useful for code checking. This information, a portion of which is illustrated in Figure 11, is provided at mesh location (IS_1,IS_2) the coordinates of which are entered on card 27b in format (2I5). At each Gauss point the following geometric data are listed:

$$J^{j}_{k} \equiv \mathrm{GBASE(J,K)},\ \Delta J^{j}_{k} \equiv \mathrm{DGBASE(J,K)},\ G_{jk} \equiv \mathrm{G(J,K)},\ G \equiv \mathrm{GTYPE},$$

$$G^{jk} \equiv \mathrm{GG(J,K)},\ \Delta\gamma_{jk} \equiv \mathrm{DGAM(J,K)},\ \text{and}\ \Delta\gamma^{j}_{k} \equiv \mathrm{DGAMMX(J,K)}.$$

This is followed by a row of printing which gives the pressures $p(\xi^{\alpha},t_n) \equiv$ P(I1,I2), $p(\xi^{\alpha},t_{n+1}) \equiv$ PPL(I1,I2) and the value of G^{33} on the loaded surface. Next, the code lists for each sublayer associated with the Gauss point the arrays of $(\tau^{j}_{k})_n \equiv$ TN(J,K) and ${}^{T}\tau^{j}_{k\ n+1} \equiv$ TR(J,K). The value of $C \equiv CZ$ is then printed as well as $\sigma_Y{}^2 \equiv$ SIGMSQ. If $C \leq 0$ the stress increment in the sublayer is elastic and $(\tau^{j}_{k})_{n+1} \equiv$ TR(J,K). For $C > 0$ the stress change is elastoplastic and the following information is printed: $A \equiv AZ, B \equiv BZ$, and the discriminant $B^2 - AC \equiv$ DISCR. This is followed by ${}^{C}\tau^{j}_{k\ n} \equiv$ TC(J,K), λ = HLAMDA and ${}^{C}\tau^{j}_{k\ n+1} \equiv$ TM(J,K). Regardless of whether the stress state is elastic or plastic the code then prints "(REVISED)DGAMMX(3,3) = ." This is the sublayer strain increment $\Delta\gamma^{3}_{3}$ consistent with the constitutive relations which is determined by use of equation (19). The corresponding value of $\Delta\gamma^{3}_{3}$ obtained by use of the strain-displacement relations is printed with the rest of such quantities in the DGAMMX(J,K) array for the Gauss point. When the preceding material has been printed for all sublayers at one Gauss point the corresponding material is listed for all the other Gauss points at the selected mesh point. At the end of this output group the value of ${}^{m}\Delta\gamma^{3}_{3} \equiv$ AGAM33 is printed.

Print option K=13 has been added which provides the array of total Gauss point mixed tensor stresses TAUF(J,K), a sample of which is shown in Figure 12. The frequency of output of this array is controlled by the value assigned to IOUT(13) on card 6a. The location(s) at which these stresses are printed are determined by entries on two cards: on card 27c the value of NUM (= number of points at which mixed tensor stresses are to be printed (and plotted)) in format (I5) and on card 27d the values of coordinate pairs IPS_1(I),IPS_2(I) in format (2I5) for I = 1,NUM.

```
ITIME= 35 TIME= .265625000E-04 I1= 3 I2= 4 IGAUSS= 1
SUBROUTINE ZETA
GBASE(1,1)= .37093325916957 38E+00 GBASE(1,2)= -.41325690877446 73E-03 GBASE(1,3)= .35550979437237E-02
GBASE(2,1)= -.36006705746059 66E-03 GBASE(2,2)= .40110469881059 9E+00 GBASE(2,3)= .77546946875773 3E-02
GBASE(3,1)= -.86970461222963 3E-02 GBASE(3,2)= -.17738543350485 7E-01 GBASE(3,3)= .99721549369890 3E+00
DGBASE(1,1)= .10970C0355264 50E-03 DGBASE(1,2)= -.30068335299866 0E-04 DGBASE(1,3)= .32061433538714 0E-03
DGBASE(2,1)= -.35205269671814 1E-04 DGBASE(2,2)= .69576202570583 6E-04 DGBASE(2,3)= .69985940042957 6E-03
DGBASE(3,1)= -.80697488641759 9E-03 DGBASE(3,2)= -.16514281418852 6E-02 DGBASE(3,3)= -.19898313970358 4E-04
G(1,1)= .14360321875659 0E+00 G(1,2)= -.27463149870235 25E-03 G(1,3)= .25693120702737 5E-03
G(2,1)= -.27463149870235 25E-03 G(2,2)= .16094524434592 4E+00 G(2,3)= .62122012477813 8E-03
G(3,1)= .25693120702737 5E-03 G(3,2)= .62122012477813 8E-03 G(3,3)= .99948290358034 8E+00
GTYPE= .43492251524492 1E+02
GG(1,1)= .69636581819023 0E+01 GG(1,2)= .11889541415162 1E-01 GG(1,3)= -.18059053965515 6E-02
GG(2,1)= .11889541415162 1E-01 GG(2,2)= .62133285236019 7E+01 GG(2,3)= -.38829782564865 6E-02
GG(3,1)= -.18059053965515 6E-02 GG(3,2)= -.38829782564865 6E-02 GG(3,3)= .10052007331792 7E+01
DGAM(1,1)= .42663367203337 0E-04 DGAM(1,2)= -.10356672355679 2E-04 DGAM(1,3)= .70841257616294 9E-05
DGAM(2,1)= -.10356672355679 2E-04 DGAM(2,2)= .33099272289206 1E-04 DGAM(2,3)= .17411663817748 4E-04
DGAM(3,1)= .70841257616294 9E-05 DGAM(3,2)= .17411863817746 7E-04 DGAM(3,3)= .14779910602335 0E-04
DGAMMX(1,1)= .29695717674718 2E-03 DGAMMX(1,2)= -.71758235197044 9E-04 DGAMMX(1,3)= .49511756277254 9E-04
DGAMMX(2,1)= -.63869667392165 7E-04 DGAMMX(2,2)= .20546590665148 8E-03 DGAMMX(2,3)= .10821246704301 0E-03
DGAMMX(3,1)= .70841371380232 4E-05 DGAMMX(3,2)= .17392597691511 3E-04 DGAMMX(3,3)= .14776373824239 2E-04
P(I1,I2)= .16594906175148 6E+06 PPL(I1,I2)= .16146641116073 8E+06 SURFACE GG(3,3)= .10052025646956 5E+01
SUBROUTINE STRESS
ISB= 1

TN(1,1)= .13563038450841 0E+06 TN(1,2)= -.20597361501095 2E+05 TN(1,3)= .19957398016636 3E+05
TN(2,1)= -.18342864037749 6E+05 TN(2,2)= .11693957404146 6E+06 TN(2,3)= .42901145563243 2E+05
TN(3,1)= .28556530707226 0E+04 TN(3,2)= .68891836810516 5E+04 TN(3,3)= -.11582094128851 4E+05
TR(1,1)= .14729079573169 4E+06 TR(1,2)= -.22279195138526 0E+05 TR(1,3)= .21117829851259 5E+05
TR(2,1)= -.19839809367253 4E+05 TR(2,2)= .12645565822188 2E+06 TR(2,3)= .45437375259563 8E+05
TR(3,1)= .30216875348950 3E+04 TR(3,2)= .72968226894464 5E+04 TR(3,3)= -.11270694306165 0E+05
  *** ISB= 1 L= 1 LC= 1 CZ= .11212156296506E+10 SIGMSQ= .23104600000000 0E+11
AZ= .60852114129941 1E+10 BZ= .98275109396702 3E+10 DISCR= .89763222334685 6E+20
TC(1,1)= .91044412013495 3E+05 TC(1,2)= -.20597361501095 2E+05 TC(1,3)= .19957398016636 3E+05
TC(2,1)= -.18342864037749 6E+05 TC(2,2)= .72353601146551 3E+05 TC(2,3)= .42901145563243 2E+05
TC(3,1)= .28556530707226 0E+04 TC(3,2)= .68891836810516 5E+04 TC(3,3)= 0.
HLAMDA= .58036678185087 1E-01
TM(1,1)= .14200688049075 1E+06 TM(1,2)= -.21083742697542 6E+05 TM(1,3)= .19959568765076 3E+05
TM(2,1)= -.18775250470028 4E+05 TM(2,2)= .12225649555631 8E+06 TM(2,3)= .42947535280566 7E+05
TM(3,1)= .28555549166098 1E+04 TM(3,2)= .68969973531637 2E+04 TM(3,3)= -.11270694306165 0E+05
(REVISED) DGAMMX(3,3)= -.32615243524054 12E-03
SUBROUTINE STRESS
ISB= 2

TN(1,1)= .13563038450841 0E+06 TN(1,2)= -.20597361501095 2E+05 TN(1,3)= .19957398016636 3E+05
TN(2,1)= -.18342864037749 6E+05 TN(2,2)= .11693957404146 6E+06 TN(2,3)= .42901145563243 2E+05
TN(3,1)= .28556530707226 0E+04 TN(3,2)= .68891836810516 5E+04 TN(3,3)= -.11582094128851 4E+05
TR(1,1)= .14729079573169 4E+06 TR(1,2)= -.22279195138526 0E+05 TR(1,3)= .21117829851259 5E+05
TR(2,1)= -.19839809367253 4E+05 TR(2,2)= .12645565822188 2E+06 TR(2,3)= .45437375259563 8E+05
TR(3,1)= .30216875348950 3E+04 TR(3,2)= .72968226894464 5E+04 TR(3,3)= -.11270694306165 0E+05
  *** ISB= 2 L= 1 LC= 1 CZ= -.92827111770366 89E+11 SIGMSQ= .16402500000000 0E+12
(REVISED) DGAMMX(3,3)= -.18726729261165 8E-03
SUBROUTINE STRESS
ISB= 3

TN(1,1)= .13563038450841 0E+06 TN(1,2)= -.20597361501095 2E+05 TN(1,3)= .19957398016636 3E+05
TN(2,1)= -.18342864037749 6E+05 TN(2,2)= .11693957404146 6E+06 TN(2,3)= .42901145563243 2E+05
TN(3,1)= .28556530707226 0E+04 TN(3,2)= .68891836810516 5E+04 TN(3,3)= -.11582094128851 4E+05
TR(1,1)= .14729079573169 4E+06 TR(1,2)= -.22279195138526 0E+05 TR(1,3)= .21117829851259 5E+05
TR(2,1)= -.19839809367253 4E+05 TR(2,2)= .12645565822188 2E+06 TR(2,3)= .45437375259563 8E+05
TR(3,1)= .30216875348950 3E+04 TR(3,2)= .72968226894464 5E+04 TR(3,3)= -.11270694306165 0E+05
  *** ISB= 3 L= 1 LC= 1 CZ= -.15624911177036 6E+13 SIGMSQ= .23685209999999 7E+13
(REVISED) DGAMMX(3,3)= -.18726729261165 8E-03
SUBROUTINE STRESS
ISB= 4

TN(1,1)= .13563038450841 0E+06 TN(1,2)= -.20597361501095 2E+05 TN(1,3)= .19957398016636 3E+05
TN(2,1)= -.18342864037749 6E+05 TN(2,2)= .11693957404146 6E+06 TN(2,3)= .42901145563243 2E+05
TN(3,1)= .28556530707226 0E+04 TN(3,2)= .68891836810516 5E+04 TN(3,3)= -.11582094128851 4E+05
TR(1,1)= .14729079573169 4E+06 TR(1,2)= -.22279195138526 0E+05 TR(1,3)= .21117829851259 5E+05
TR(2,1)= -.19839809367253 4E+05 TR(2,2)= .12645565822188 2E+06 TR(2,3)= .45437375259563 8E+05
TR(3,1)= .30216875348950 3E+04 TR(3,2)= .72968226894464 5E+04 TR(3,3)= -.11270694306165 0E+05
  *** ISB= 4 L= 1 LC= 1 CZ= -.98138771177037 5E+13 SIGMSQ= .14745600000000 0E+14
(REVISED) DGAMMX(3,3)= -.18726729261165 8E-03
ITIME= 35 TIME= .265625000E-04 I1= 3 I2= 4 IGAUSS= 2
SUBROUTINE ZETA
GBASE(1,1)= .37820531892102 6E+00 GBASE(1,2)= -.18699037712557 2E-03 GBASE(1,3)= .35754576051022 6E-02
GBASE(2,1)= -.16895711120164 4E-03 GBASE(2,2)= .40054662638434 6E+00 GBASE(2,3)= .77426773770687 4E-02


SUBROUTINE STRESS
ISB= 4

TN(1,1)= -.14810755308420 6E+06 TN(1,2)= .19588677611103 2E+05 TN(1,3)= .25825522311495 5E+05
TN(2,1)= .17451832943619 3E+05 TN(2,2)= -.13019296235269 0E+06 TN(2,3)= .54100339913526 5E+05
TN(3,1)= .36898301049738 6E+04 TN(3,2)= .86826998159050 1E+04 TN(3,3)= -.15523032835257 9E+06
TR(1,1)= -.15582946030589 9E+06 TR(1,2)= .21665651178241 9E+05 TR(1,3)= .27336721457602 8E+05
TR(2,1)= .19302889441605 0E+05 TR(2,2)= -.13529015747807 4E+06 TR(2,3)= .56821547830329 0E+05
TR(3,1)= .39065541181935 7E+04 TR(3,2)= .91213801873077 5E+04 TR(3,3)= -.15105675695981 0E+06
  *** ISB= 4 L= 1 LC= 1 CZ= -.98280843435354 4E+13 SIGMSQ= .14745600000000 0E+14
(REVISED) DGAMMX(3,3)= .25876586831677E-03
ACAM33= .12587413707904E-04
```

Figure 11. Sample of Geometric and Stress Output Data

```
TIME=    .293750E-03   ITIME =  500 I1=  2 I2=  3 GAUSS PT. =  1  TAUSPH =      328.0
                                                  MIXED TENSOR STRESSES
                                   TAUF(1,1)=  -94615.2 TAUF(1,2)=          .0 TAUF(1,3)=        -.0
                                   TAUF(2,1)=        .0 TAUF(2,2)=     95594.7 TAUF(2,3)=     4127.1
                                   TAUF(3,1)=        .0 TAUF(3,2)=      1349.4 TAUF(3,3)=         .5
TIME=    .293750E-03   ITIME =  500 I1=  2 I2=  3 GAUSS PT. =  2  TAUSPH =    35948.2
                                                  MIXED TENSOR STRESSES
                                   TAUF(1,1)=  -14415.4 TAUF(1,2)=          .0 TAUF(1,3)=        -.0
                                   TAUF(2,1)=        .0 TAUF(2,2)=    126257.6 TAUF(2,3)=     -672.1
                                   TAUF(3,1)=        .0 TAUF(3,2)=       513.4 TAUF(3,3)= [illegible]
TIME=    .293750E-03   ITIME =  500 I1=  2 I2=  3 GAUSS PT. =  3  TAUSPH =    32231.5
                                                  MIXED TENSOR STRESSES
                                   TAUF(1,1)=  135169.7 TAUF(1,2)=         -.0 TAUF(1,3)=         .0
                                   TAUF(2,1)=       -.0 TAUF(2,2)=    -38480.2 TAUF(2,3)=    -6263.7
                                   TAUF(3,1)=       -.0 TAUF(3,2)=     -1243.8 TAUF(3,3)=        5.0
TIME=    .293750E-03   ITIME =  500 I1=  2 I2=  3 GAUSS PT. =  4  TAUSPH =     5605.5
                                                  MIXED TENSOR STRESSES
                                   TAUF(1,1)=  133779.5 TAUF(1,2)=         -.0 TAUF(1,3)=        -.0
                                   TAUF(2,1)=       -.0 TAUF(2,2)=    -83361.0 TAUF(2,3)=   -10016.4
                                   TAUF(3,1)=       -.0 TAUF(3,2)=     -1341.3 TAUF(3,3)=        7.0
```

Figure 12. Sample Printout of Gauss Point Stresses

TIME= .293750E-03   ITIME= 500
SUBDIVISIONS OF TIME INCREMENT IN STRESS

LMAT(I1,I2 1) LAYER 1 GAUSS PT. 1 SUBLAYER 1

I1	I2= 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	2	2	2	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	2	2	2	3	2	2	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	2	2	2	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	2	3	2	2	2	0	2	1	0	0	0	0	0	0	0	0	0	0	0
6	0	0	2	0	2	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	2	2	2	0	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	0	1	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13. Sample Printout of Elastoplastic Activity Matrix

Print option K=14 provides the matrix of integer values L discussed in Section B of Chapter III (see Figure 13). These integers indicate the level of elastoplastic activity currently taking place at each mesh location: zero indicates elastic behavior, one signifies normal plastic behavior, and any integer ≥ 2 specifies rapid plastic flow requiring the code to subdivide the time step into L equal substeps for purposes of stress evaluation. The code will provide an L matrix for each sublayer at each Gauss point at time intervals determined by the value assigned to IOUT(14).

E. Additions to Plotted Output

The PETROS 4 program had already been altered to couple with the REPSIL plotting program (Appendix D of Reference 7). With this plotting package one obtains isometric plots of the deformed shell surface at selected time intervals, two dimensional plots of displacement vs time, load vs time, and surface strains vs time as well as an energy balance diagram. The plot of pressure loading vs time is generated for mesh location (IP_1, IP_2), the coordinates of which are entered on card 27a in format (2I5).

The following plotted output has been added:

1. The cartesian components of $\overline{D}$, i.e., $Y^{(2)k}$ are plotted vs time at the same mesh location already selected for $Y^{(1)k}$.

2. Mixed tensor stress components τ_k^j vs time (see Figures 1-5) are plotted for each Gauss point through the thickness for the mesh point location selected for print option K=13.

3. Also, for the same location, a plot of the through-thickness strain component γ_3^3 vs time has been added.

VI. CONCLUDING REMARKS

The modifications to the PETROS 4 code discussed in the foregoing text have resulted in an improved version which is suitable for use even in rather exceptional applications such as those cited in the Introduction, where the hydrostatic component of stress is a significant fraction of the largest principal stress. The concept of prescribed through-thickness normal stress is considered to be a novel approximate procedure for taking account of this stress component in elastoplastic stress evaluations. The problem of unstable growth of through-thickness strain γ_3^3 has been successfully circumvented with the introduction of the INORML = 3 (recycling) option. It is believed that the SHEAR option with the ISTRES = 4, INORML = 3 combination will satisfactorily treat the elastoplastic response of panels subjected to blast from nearby explosive charges and serve as a point-of-departure for studies of panel rupture.

The difficulty with the ISTRES = 0, INORML = 0 combination cited in Chapter II persists and is not alleviated by the addition of surface traction terms and use of recycling. In retrospect, it is concluded that the ISTRES = 0 option will not, in general, give a satisfactory representation for stresses in thin and moderately thick shells since there is nothing in the basic PETROS 4 formulation to enforce the stress boundary conditions on the two shell surfaces (or more significantly, at the Gauss points closest to the surfaces). By contrast the ISTRES = 4 option does satisfy the surface boundary condition on the normal stress and by use of a constrained three-dimensional constitutive formulation provides elastoplastic stresses which appear to be realistic.

The INORML = 2 option was intended to take account of an average thickness change by modifying the $Y^{(2)j}$ variables at the <u>next</u> time step. While the problem with this option (shown in Figure 5) was not resolved the subject appears moot since the new INORML = 3 option takes account of $\Delta\gamma_3^3$ changes in the <u>current</u> time step. One remaining issue which deserves further study is correction or improvement of the formulation for the plastic work.

The cited modifications to the PETROS 4 program have affected only a few subroutines of this rather lengthy code; a listing of these revised subroutines is provided in the Appendix. Consideration was given to inclusion of an application of the modified PETROS 4 code in this report but, owing to the complexity of such results, it is preferred to present these as a separate document.

REFERENCES

1. S. D. Pirotin, L. Morino, E. A. Witmer, and J. W. Leech, "Finite-Difference Analysis for Predicting Large Elastic-Plastic Transient Deformations of Variable-Thickness Kirchhoff, Soft Bonded Thin, and Transverse-Shear Deformable Thicker Shells," US Army Ballistic Research Laboratory Contract Report No. 315, September 1976. AD B 013924L

2. S. D. Pirotin, B. A. Berg, and E. A. Witmer, "PETROS 4: New Developments and Program Manual for the Finite-Difference Calculation of Large Elastic-Plastic, and/or Viscoelastic Transient Deformations of Multilayer Variable-Thickness (1) Thin Hard-Bonded, (2) Moderately-Thick Hard-Bonded, or (3) Thin Soft-Bonded Shells," US Army Ballistic Research Laboratory Contract Report No. 316, September 1976. AD B 014253L

3. H. F. Bohnenblust, and P. Duwez, "Some Properties of a Mechanical Model of Plasticity," Journal of Applied Mechanics, Vol. 15, No. 3, September 1948, pp. 222-225.

4. G. N. White, Jr., "Application of the Theory of Perfectly Plastic Solids to Stress Analysis of Strain Hardening Solid," Graduate Div. of Applied Math., Brown University Tech Report 51, August 1950.

5. J. F. Besseling, "A Theory of Plastic Flow for Anisotropic Hardening in Plastic Deformation of an Initially Isotropic Material," Report 5410, National Aeronautical Research Institute, Amsterdam, The Netherlands, 1953.

6. N. J. Huffington, Jr., "Numerical Analysis of Elastoplastic Stresses," US Army Ballistic Research Laboratory Memorandum Report No. 2006, September 1969. AD 861688

7. J. M. Santiago, H. L. Wisniewski, and N. J. Huffington, Jr., "A User's Manual for the REPSIL Code," US Army Ballistic Research Laboratory Report No. 1744, October 1974. AD A 003176.

NOMENCLATURE

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
	$\bar{A}_\alpha$	Deformed reference surface base vectors
A(LA,LB)		Covariant components of metric tensor of deformed reference surface
ABSCIS (IGAUSS,NGAUSL)		Location of IGAUSSth of the N Gaussian stations in a particular layer; interval is -1 to +1.
ACC(J,I1,I2)	$\ddot{Y}^{(1)j}$	Acceleration components
ALPHA		Coefficient of linear thermal expansion
AGAM33	$\overset{m}{\Delta}\gamma_3^3$	Gaussian average of SGAM33
ANUM		Status of material (in FMAT)
AVEG33	G^{33}	Gaussian average G^{33}
AVIS (ILAYER)		Viscoelastic coefficient of ILAYER
AZ	A	Coefficient in quadratic equation
B(LA,LB)		Covariant components of deformed reference surface curvature tensor
BM(LA,LB)		Mixed curvature tensor components of deformed reference surface
BSTIV(ILAYER)		Elastic modulus coefficient of ILAYER
BZ	B	Coefficient in quadratic equation
CAPQ1(LA,I1,I2) CAPQ2(LA,I1,I2) CAPQ3(LA,I1,I2)		Generalized force resultant tensor
CAP2Q1(LA,I1,I2) CAP2Q2(LA,I1,I2) CAP2Q3(LA,I1,I2)		Generalized force in EQUIL2
COEFF(ISB)		Weighting factors of the mechanical sublayer model

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
CONST		Strain-rate sensitivity parameter
CS(J,LA)		See ZETA 158-163
CX(I,J)		See ZETA 358-375
CZ	$C, \overset{T}{\phi}_{n+1}$	Coefficient in quadratic equation
C1		Viscous damping parameter
	$\overline{D}(\xi^\alpha, t)$	Non-Kirchhoff displacement field
	D^α, D^3	Tangential and normal components of $\overline{D}$ in basis $\overline{G}_i$
D(J,I1,I2)	$\Delta Y^{(1)j}$	Incremental change in $Y^{(1)j}$
DA(LA,LB)		Incremental change in the covariant components of the metric tensor associated with the deformed midsurface of the shell
DB(LA,LB)		Incremental change in the corresponding curvature tensor
DD(J,LA)	$\dfrac{\partial \Delta Y^{(1)j}}{\partial \xi^\alpha}$	
DD2(J,LA)	$\dfrac{\partial \Delta Y^{(2)j}}{\partial \xi^\alpha}$	
DEL		See ZETA 146,156
DELBAR	η	See ZETA 409
DELNOR		See EQUIL2 174
DELSN1		See EQUIL2 176,180
DELSN2		See EQUIL2 177,181
DELSN3		See EQUIL2 178,182
DELTA(I,J)	δ^i_j	Kronecker delta

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
DELTAP		Previous time increment
DELTAT	Δt	Time increment
DGAM(I,J)	$\Delta\gamma_{ij}$	Covariant components of strain increment
DGAMAT		See STRESS 263
DGAMA3		Average $\Delta\gamma_3^3$
DGAMMA		See STRESS 94, 95, 96
DGAMMX(I,J)	$\Delta\gamma_i^j$	Mixed components of the incremental strain tensor
DGAM33	$\Delta\gamma_3^3$	Mixed component of incremental strain tensor at Gauss point
DGBASE(I,J)	ΔJ_i^j	Cartesian components of base vector increment
DGTEMP		See STRESS 93
DGM33	$\Delta\gamma_3^3$	Gaussian average $\Delta\gamma_3^3$
DGOG		See ZETA 408
DISCR		Discriminant of quadratic equation
DJR		Saved value of D from previous time step
DN(J)	ΔN^j	Incremental change in component of surface normal
DTAU33		See STRESS 123
DTEMP		Temperature increment
DTM(I,J)		Incremental change in stress
DUM		Intermediate variable
DX1		Increment in ξ^1 coordinate
DX2		Increment in ξ^2 coordinate

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
DZ(I)	$\frac{\partial z}{\partial \xi^{\alpha}}, \frac{\partial z}{\partial \zeta}$	
DZA1(I1,I2)	$\frac{\partial z^{(A)}}{\partial \xi^{1}}$	
DZA2(I1,I2)	$\frac{\partial z^{(A)}}{\partial \xi^{2}}$	
DZB1(I1,I2)	$\frac{\partial z^{(B)}}{\partial \xi^{1}}$	
DZB2(I1,I2)	$\frac{\partial z^{(B)}}{\partial \xi^{2}}$	
D2(J,I1,I2)	$\Delta Y^{(2)j}$	Incremental change in $Y^{(2)j}$
D33S		See ZETA 297
EE	E	Young's modulus
EEP		See STRESS 102
EL		Number of subdivisions of time step
EPSL1(I1,I2) EPSL2(I1,I2)		Normal strain components on lower surface
EPSU1(I1,I2) EPSU2(I1,I2)		Normal strain components on upper surface
ES	E	Young's modulus
ETERM1		See STRESS 195
ETERM2		See STRESS 196
E1(J,I1,I2)	$\tilde{E}^{j}_{(1)}$	Surface force term for EQUIL
E2(J,I1,I2)	$\tilde{E}^{j}_{(2)}$	Surface force term for EQUIL2
FACTOR		Coefficient of strain-rate sensitivity
FMAS11(I1,I2) FMAS22(I1,I2) FMAS23(I1,I2)		Generalized masses

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
FMAT		Material status array
FORCES(J)		Component of externally-applied force per unit area in j-direction
FORCEZ(J)		Convenient grouping of force components
	$\overline{G}_3$	Covariant basis vector component of deformed shell in direction of normal
G(I,J)	G_{ij}	Covariant components of the metric tensor of the deformed surface
GAMMAL(I1,I2)		Shear strain component on lower surface
GAMMAU(I1,I2)		Shear strain component on upper surface
GBASE(I,J)	J_i^j	Cartesian components of the base vector $\overline{G}_i$ in j-direction
GBTN	$J_3^j N^j$	See ZETA 225
GG(I,J)	G^{ij}	Contravariant components of the metric tensor of the deformed surface
GTYPE	G	Metric determinant
HLAMDA	λ	Plasticity parameter
HM(LA,J)		Contravariant components of the relative moment-resultant tensor
HM1(LA,I1,I2) HM2(LA,I1,I2)		Storage of components of HM
HN(LA,J)		Contravariant components of the relative stress-resultant tensor
HNU	ν	Poisson's ratio
HNUP		See STRESS 104,105
HNUPP		See STRESS 103,106

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
HTERM		See STRESS 241
	$\bar{i}_k$	Cartesian unit vector in k-direction
I		Component index
ICOUNT		Output control counter
IC1 IC2		Mesh indices selected by user at which specific output is desired
IFRACT		Failure model selector
IGAUSS	IGAUSS	Gauss point index
IGMAX(I,J)		Gauss point index where maximum value of stress component occurs
IGMIN(I,J)		Gauss point index where minimum value of stress component occurs
IGO		Selector for calculation or output of max/min stresses
III		Output control index
IJ		Number of components
ILAYER		Layer index
INORML	INORML	Control number for options regarding $Y^{(2)}$ modification
IOUT(K)		Printout indicator
IPLAST(I1,I2)		Plasticity indicator
IPS1(I) IPS2(I)		Coordinates of locations at which output of Gauss point mixed tensor stresses is desired
IRY1,IRY2, IRY3,IRY4		Indices corresponding to the limits of the complete finite difference grid
ISB		Sublayer index
ISTRES	ISTRES	Plasticity model control

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
ISTREZ		Plasticity model control
ISUBL		Sublayer index
IS1(I) IS2(I)		Coordinates of locations at which output of geometric and stress variables is desired (IOUT(12))
ITIM(I,J)		Time cycle of maximum value of stress component
ITIME		Current cycle number
ITIMEF		Final cycle number
ITIMEP		ITIME-1
ITIMM(I,J)		Time cycle of minimum value of stress component
IV		Component index
IZ		Gauss layer counter
IZZ		Upper/lower surface selector
I1	I1	Mesh point index
I1M		I1-1
I1MAX(I,J)		I1 location where maximum value of stress component occurs
I1MIN(I,J)		I1 location where minimum value of stress component occurs
I1P		I1+1
I2	I2	Mesh point index
I2M		I2-1
I2MAX(I,J)		I2 location where maximum value of stress component occurs
I2MIN(I,J)		I2 location where minimum value of stress component occurs

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
I2P		I2+1
J		Component index
JD1		Number of stress memory locations in ξ^1 -direction
JD2		Number of stress memory locations in ξ^2 -direction
JD3		Number of stress memory locations in ζ -direction
JV		Component index
	K	Bulk modulus = $E/(3(1-2\nu))$
K		Component index
K1,K2,K3,K4		Boundary condition control indices on the four boundary lines
KF		Number of print options available
L	L	Number of subdivisions of time increment
LA,LB		Component indices
LC		Counter for time increment subdivision
LEN		See STRESS 73
LI1I2		Input to LMAT
LL		Index of mesh points for IOUT(13)
LMAT(I1,I2,IZ)		Plasticity activity arrays
LS		Component index
LZ		See STRESS 444
MAX		Integer controlling output heading
MAUXIL		Controls max/min stress output
MIN		Integer controlling output heading

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
MPHYS		PHYSIC control variable
MTEMPE		Index for temperature effects
	$\overline{N}$	Deformed reference surface normal
	$\overline{n}$	Undeformed reference surface normal
NGAUSL		Number of Gauss stations in layer
NGAUSS(ILAYER)		Number of Gauss stations in layer = ILAYER
NLAYER		Number of layers
NMESH1		Number of meshes in ξ^1 -direction
NMESH2		Number of meshes in ξ^2 -direction
NSBL		Number of sublayers
NSUBL(ILAYER)		Number of sublayers in i-th layer
NUM		Number of mesh points at which IOUT(13) is desired
P(I1,I2)	p	Pressure
	p_o	Peak value of pressure
PAR		See ZETA 304
PARSQ		See ZETA 305
PARSQZ		See ZETA 376
PGAM33	γ_3^3	See ZETA 413
PPL(I1,I2)		Pressure at next time step
PRSQD1		See ZETA 319
PRSQD2		See ZETA 320
PRSQD3		See ZETA 321
PRSQZZ		See ZETA 343

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
QCZ		Logical variable for time step subdivision
QIRCH		Logical variable for Kirchhoff shell theory
QM		Logical variable for max/min stress calculation
QPRINT(20)		Printout indicator
QQQ2,QQQ3,QQQ4		Logical variables used in STRESS for defining coefficients of constitutive functions
QQ1,QQ2		Logical variables used in EQUIL to avoid calculations at boundary points
Q11		Logical variable for maximum or minimum stress selection
QSHEAR		Logical variable for SHEAR option
QSTRES		Logical variable for stress erasure
	$\bar{r}_o$	Undeformed shell reference surface position vector
SGAM33		See ZETA 398
SIGMA(ISB)		Uniaxial yield stress of the ISBth sublayer
	σ_Y	Static uniaxial yield stress of the material
SIGN		$\pm 1.$
SN(J,I1,I2)	N^j	Components of the surface normal
SQRG	$\sqrt{G}$	$\sqrt{G}$
STRESL(J,I1,I2) STRESP(LA,I1,I2) STRESQ(LA,I1,I2)		Generalized forces calculated in ZETA
SUMG		See ZETA 396
SURFGG		G^{33} at shell surface
TAU(I,J)	τ^{ij}	Contravariant stress components at Gauss points

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
TAUC	$\overset{C}{\tau}{}^{m}_{m}$	Trace of the corrector stress tensor
TAUF(I,J)	τ^{i}_{j}	Mixed tensor stress components at Gauss points
TAUM	$(\tau^{m}_{m})_{n+1}$	Trace of the new mixed stress tensor
TAUMAX(I,J)		Maximum value of the stress component
TAUMIN(I,J)		Minimum value of the stress component
TAUP(LZ,I,J)		Storage of TAUF stresses
TAUSPH		Hydrostatic stress
TAUSPL(LL,IGAUSS)		Storage of TAUSPH
TAUSUM		See ZETA 400
TAUT	$\overset{T}{\tau}{}^{m}_{m}$	Trace of the trial stress tensor
TAU11(I1,I2,IZ) TAU12(I1,I2,IZ) TAU13(I1,I2,IZ) TAU21(I1,I2,IZ) TAU22(I1,I2,IZ) TAU23(I1,I2,IZ) TAU31(I1,I2,IZ) TAU32(I1,I2,IZ) TAU33(I1,I2,IZ)		Storage of mixed tensor sublayer stresses
TC(I,J)	$\overset{C}{\tau}{}^{i}_{j}$	Mixed tensor corrector stress components
THIC		See ZETA 201
THICKN	h	Shell thickness
THICKZ		See ZETA 200
THIKZ		See ZETA 151
TIM(I,J)		Time of maximum value of stress component

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
TIME	t, t_n	Time
TIMM		Time of minimum value of stress component
TM(I,J)	$(\tau^{i}_{j})_{n+1}$	New mixed tensor sublayer stresses
TN(I,J)	$(\tau^{i}_{j})_{n}$	Previous mixed tensor sublayer stresses
TR(I,J)	$\overset{T}{(\tau^{i}_{j})}_{n+1}$	Trial mixed tensor sublayer stresses
T33		See STRESS 121
T33PL		See STRESS 122
	$\bar{u}$	Displacement field vector
	$\bar{u}_o$	Displacement vector of points on the reference surface
	v^{α}, w	Components of $\bar{u}_o$ in surface -normal directions
WEIGHT (IGAUSS, NGAUSL)		Gaussian weighting factors
Y(J,I1,I2)	$Y^{(1)j}$	Rectangular cartesian coordinates of mesh point (I1,I2)
YLDFAC	YLDFAC	Factor controlling subdivision of time step
YY(J,LA)	$\dfrac{\partial Y^{(1)j}}{\partial \xi^{\alpha}}$	
YYU(J,LA)	$Y^{(1)\alpha j}$	
YY2(J,LA)	$\dfrac{\partial Y^{(2)j}}{\partial \xi^{\alpha}}$	
Y2(J,I1,I2)	$Y^{(2)j}$	Rectangular cartesian components of $\overline{D}$
Y2DOT2(J,I1,I2)		Acceleration of $Y^{(2)j}$
Y3ACEL(J)		Rectangular cartesian components of $\ddot{\overline{N}}$

FORTRAN NAME	TEXT SYMBOL	DESCRIPTION
Z	ζ	Distance from reference surface
	ζ_g	Distance of Gauss point from reference surface
ZA(I1,I2)		ζ location of interface of upper and middle layer
ZB(I1,I2)		ζ location at interface of lower and middle layer
ZCEN		See ZETA 152
ZCENTR		Value of ζ at the center of a given layer
ZZ		Displacement field parameter
ZZCEN		See ZETA 155
	γ_{33}	Covariant strain component in through-thickness direction
	ξ^α	Curvilinear coordinates of particles on the reference surface
	${}^{T}\Delta\tau^{i}_{j}$	Mixed tensor trial stress increment
	ϕ_n	Yield function
	$()_n$	Quantity evaluated at t_n
	$()_{n+1}$	Quantity evaluated at t_{n+1}
	$(^m)$	Modified quantity
	INT [ ]	Integer part of [ ]

APPENDIX A

Listing of Significantly Affected Subroutines

APPENDIX A

Listing of Significantly Affected Subroutines

```
      SUBROUTINE AUXIL(INDEXX)
C
C
      IMPLICIT LOGICAL(Q)
C
      COMMON /CARTE/ YTEST,YNEW,YSAVE
      COMMON /CARTEL/ Y(3,20,20),D(3,20,20),Y2(3,20,20),D2(3,20,20)
      LEVEL 2,Y,D,Y2,D2
C
      COMMON /CTIME/ AUX(20),TIME,DELTAT,TIMEF,ITIME,ITIMEF,IAUX(20),
     * IOUT(20),QPRINT(20)
      COMMON /CTIMEL/ IPLAST(20,20),P(20,20),PPL(20,20)
      LEVEL 2,IPLAST,P,PPL
C
      COMMON /INDEX/ NREAD,NWRITE,NPUNCH,NMESH1,NMESH2,N1,N2,NZ,N1M,N2M,
     * N1MM,N2MM,I1,I2,IZ,I1ZERO,I2ZERO,IRY1,IRY2,IRY3,IRY4,ISTR1,ISTR2,
     * ISTR3,ISTR4,IC1,IC2,ID1,ID2,IP1,IP2,IS1,IS2,K1,K2,K3,K4,KRUN,
     * KZSTOP,KYTEST,IDIR,I1TEST,I2TEST,KINITL
C
      COMMON /QLOGIC/ QAUX(20),QZETA,QSTRES,QPLAST,QSENS1,QEQUIL,
     * QDIAGN,QINGEO,QINVEL,QLOAD,QMATPR,QTHIKL,QTEMPE,QSPTEM,QAUXI1,
     * QAUXI2,QSPLOA,QIMPUL,QSHARP,QPESO,QIRCH,QSHEAR
C
      COMMON /FRAC/ TAUF(3,3),TAUSPH,NUM,IPS1(10),IPS2(10)
C
      II=INDEXX
      IF(INDEXX .GE. 4) II=4
      IF(INDEXX .EQ. 11) II=5
      GOTO(1001,1002,1003,1004,1005),II
C
C
 1001 IF(I1.EQ.IC1.AND.I2.EQ.IC2) CALL PRINT(1)
      RETURN
C
C
 1002 CALL PRINT(2)
  403 FORMAT(2I5,3E15.6)
      III=ITIME/IOUT(9)*IOUT(9)-ITIME
      IF(III.NE.0.AND..NOT.QPRINT(9)) GO TO 409
      WRITE(NWRITE,401) ITIME
  401 FORMAT(////9X,"CIRCUMFERENTIAL POSITIONS AT ITIME=",I5//"    I1   I
     *2",8X,"Y1",13X,"Y2",13X,"Y3")
      IF(.NOT.QIRCH) WRITE(NWRITE,4011)
 4011 FORMAT("+",64X,"Y2(1)",10X,"Y2(2)",10X,"Y2(3)")
      DO 402 I1=ISTR1,ISTR3
      WRITE(NWRITE,403) I1,IC2,Y(1,I1,IC2),Y(2,I1,IC2),Y(3,I1,IC2)
      IF (QIRCH) GO TO 402
      WRITE(NWRITE,1403) Y2(1,I1,IC2),Y2(2,I1,IC2),Y2(3,I1,IC2)
 1403 FORMAT("+",59X,3E15.6)
  402 CONTINUE
C
      WRITE(NWRITE,411) ITIME
  411 FORMAT(////12X,"CROWN POSITIONS AT ITIME=",I5//"    I1    I2",
     *8X,"Y1",13X,"Y2",13X,"Y3")
      IF(.NOT.QIRCH) WRITE(NWRITE,4011)
      DO 412 I2=ISTR2,ISTR4
      WRITE(NWRITE,403) IC1,I2,Y(1,IC1,I2),Y(2,IC1,I2),Y(3,IC1,I2)
      IF (QIRCH) GO TO 412
      WRITE(NWRITE,1403) Y2(1,IC1,I2),Y2(2,IC1,I2),Y2(3,IC1,I2)
  412 CONTINUE
  409 CONTINUE
```

```
C
      III=ITIME/IOUT(10)*IOUT(10)-ITIME
      IF(III.NE.0.AND..NOT.QPRINT(10)) GO TO 429
C
      WRITE(NWRITE,466)ITIME,TIME,IC1,IC2,Y(1,IC1,IC2),Y(2,IC1,IC2),
     * Y(3,IC1,IC2)
  466 FORMAT(" ITIME=",I5,"    TIME=",E13.6/"    POSITION OF DESIRED POINT
     * (I1=",I2,",I2=",I2,") IS    Y(1) =",E13.6,"  Y(2) =",E13.6,
     *"  Y(3) =",E13.6)
      IF (QIRCH) GO TO 429
      WRITE(NWRITE,1466) Y2(1,IC1,IC2),Y2(2,IC1,IC2),Y2(3,IC1,IC2)
 1466 FORMAT(48X,"Y2(1)=",E13.6,"  Y2(2)=",E13.6,"  Y2(3)=",E13.6)
  429 CONTINUE
C
      RETURN
C
C
 1003 IF(I1.EQ.IC1.AND.I2.EQ.IC2) CALL PRINT(3)
      RETURN
 1004 IF(I1.EQ.IS1 .AND. I2.EQ.IS2) CALL PRINT(INDEXX)
      RETURN
C             MIXED TENSOR STRESSES CHECK
 1005 III=ITIME/IOUT(13)*IOUT(13)-ITIME
      IF(III.NE.0 .AND. .NOT. QPRINT(13))GOTO 1020
      DO 1010 L=1,NUM
      IF(I1.EQ.IPS1(L) .AND. I2.EQ.IPS2(L))GOTO 1015
 1010 CONTINUE
      GOTO 1020
 1015 CALL PRINT (INDEXX)
 1020 RETURN
      END
```

```
      SUBROUTINE EQUIL
C
C
C     EVALUATE THE DISPLACEMENT INCREMENTS AND FROM THEM EVALUATE THE
C     NEW POSITIONS
C
      IMPLICIT LOGICAL(Q)
C
      COMMON /CARTE/ YTEST,YNEW,YSAVE
      COMMON /CARTEL/ Y(3,20,20),D(3,20,20),Y2(3,20,20),D2(3,20,20)
      LEVEL 2,Y,D,Y2,D2
C
      COMMON /CTIME/ AUX(20),TIME,DELTAT,TIMEF,ITIME,ITIMEF,IAUX(20),
     * IOUT(20),QPRINT(20)
      COMMON /CTIMEL/ IPLAST(20,20),P(20,20),PPL(20,20)
      LEVEL 2,IPLAST,P,PPL
C
      COMMON/CTIMER/ITIMEC,ITIMER,DELTAP,DELX,OHR,UNH,HEE
     * ,TKEEP,HTHIK,QFINIS,QFIND,TSTART,YSTART,YDOTF
     * ,ES,BSTIV(4),NSTIV(4)
C
      COMMON /DAMP/ MDAMP,DAMPF,DFACT,TDAMP,TOTKP,C1
C
      COMMON /FORCB/ FORCEZ(3)
C
      COMMON /INDEX/ NREAD,NWRITE,NPUNCH,NMESH1,NMESH2,N1,N2,NZ,N1M,N2M,
     * N1MM,N2MM,I1,I2,IZ,I1ZERO,I2ZERO,IRY1,IRY2,IRY3,IRY4,ISTR1,ISTR2,
     * ISTR3,ISTR4,IC1,IC2,ID1,ID2,IP1,IP2,IS1,IS2,K1,K2,K3,K4,KRUN,
     * KZSTOP,KYTEST,IDIR,I1TEST,I2TEST,KINITL
C
      COMMON /PUSH/ FORCES(3),VELOC(3),RATIO,RATIOM,DX1,DX2,TEMP,DTEMP,
     * FSPACE,TSPACE,FINCND,FSTOP,TSTOP,THCOEF
      COMMON /PUSHL/ SQRAT(20,20),SQRAZ(20,20),FMAS11(20,20),
     * FMAS22(20,20),FMAS23(20,20),FMAS33(20,20)
      LEVEL 2,SQRAT,SQRAZ,FMAS11,FMAS22,FMAS23,FMAS33
C
      COMMON /QBCOND/ Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11,Q12,Q13,Q14,Q15
     * ,QFREE1,QFREE2,QFREE3,QFREE4,QFCORN,QQ1,QQ2                      ,Q
     * TRAC1,QTRAC2,QTRAC3,QTRAC4
C
      COMMON /SURNOM/ SNPR(3)
      COMMON /SURNOL/ SN(3,20,20)
      LEVEL 2,SN
C
      COMMON /S2/ STRESE(3)
      COMMON /S2L/ STRESL(3,20,20),STRESQ(2,20,20),STRESP(2,20,20)
      LEVEL 2,STRESL,STRESQ,STRESP
C
      COMMON /TNCOMP/ HM1(2,20,20),HM2(2,20,20),CAPQ1(2,20,20),
     * CAPQ2(2,20,20),CAPQ3(2,20,20),CAP2Q1(2,20,20),
     * CAP2Q2(2,20,20),CAP2Q3(2,20,20)
      LEVEL 2,HM1,HM2,CAPQ1,CAPQ2,CAPQ3,CAP2Q1,CAP2Q2,CAP2Q3
C
      COMMON /VELS/ Y3ACEL(3),Y3DOT(3),Y3ACPO(4,3)
      COMMON /VELSL/ VELO(3,20,20),ACC(3,20,20)
      LEVEL 2,VELO,ACC
C
      COMMON /SURFOR/ E1(3,20,20),E2(3,20,20)
      LEVEL 2,E1,E2
C
C
      TERMT=DELTAT/DELTAP
C
      QQ1=I1.EQ.IRY1.OR.I1.EQ.IRY3
      QQ2=I2.EQ.IRY2.OR.I2.EQ.IRY4
C
      I1P=I1+1
      I1M=I1-1
      I2P=I2+1
      I2M=I2-1
C
      L1=ISIGN(1,N1-2*I1)
      L2=ISIGN(1,N2-2*I2)
C
      TERM=1./FMAS11(I1,I2)
      TERM2=DELTAT*DELTAT
      DO 298 J=1,3
C
```

```
C               DISPLACEMENTS FOR VISCOUS DAMPING
      DJR=D(J,I1,I2)
C
C     NEXT STATEMENTS AVOID IMPERFECTIONS DUE TO ROUND-OFF ERRORS
C
      IF(K1.EQ.7.AND.J.EQ.1) GO TO 298
      IF(K2.EQ.7.AND.J.EQ.2) GO TO 298
      IF(K1.EQ.5.AND.I1.EQ.2.AND.J.EQ.1) GO TO 298
      IF(K2.EQ.5.AND.I2.EQ.2.AND.J.EQ.2) GO TO 298
      IF(K3.EQ.5.AND.I1.EQ.N1M.AND.J.EQ.1) GO TO 298
      IF(K4.EQ.5.AND.I2.EQ.N2M.AND.J.EQ.2) GO TO 298
      IF(K1.EQ.6.AND.J.EQ.1) GO TO 298
      IF(K2.EQ.6.AND.J.EQ.2) GO TO 298
C
      DERIV=0.
C
      DX=DX1
      DXFACT=.5/DX
C
      IF(QQ1) GO TO 111
C
      GO TO (102,104,106), J
  102 DERIV=DERIV+DXFACT*(CAPQ1(1,I1P,I2)-CAPQ1(1,I1M,I2))
      GO TO 119
  104 DERIV=DERIV+DXFACT*(CAPQ2(1,I1P,I2)-CAPQ2(1,I1M,I2))
      GO TO 119
  106 DERIV=DERIV+DXFACT*(CAPQ3(1,I1P,I2)-CAPQ3(1,I1M,I2))
      GO TO 119
C
C
  111 DUM=DXFACT*L1
      GO TO (112,114,116), J
  112 DERIV=DERIV+DUM*(-3.*CAPQ1(1,I1,I2)+4.*CAPQ1(1,I1+L1,I2)-CAPQ1(1,I
     * 1+2*L1,I2))
      GO TO 119
  114 DERIV=DERIV+DUM*(-3.*CAPQ2(1,I1,I2)+4.*CAPQ2(1,I1+L1,I2)-CAPQ2(1,I
     * 1+2*L1,I2))
      GO TO 119
  116 DERIV=DERIV+DUM*(-3.*CAPQ3(1,I1,I2)+4.*CAPQ3(1,I1+L1,I2)-CAPQ3(1,I
     * 1+2*L1,I2))
  119 CONTINUE
C
C
C
      DX=DX2
      DXFACT=.5/DX
C
      IF(QQ2) GO TO 131
C
      GO TO (122,124,126), J
  122 DERIV=DERIV+DXFACT*(CAPQ1(2,I1,I2P)-CAPQ1(2,I1,I2M))
      GO TO 139
  124 DERIV=DERIV+DXFACT*(CAPQ2(2,I1,I2P)-CAPQ2(2,I1,I2M))
      GO TO 139
  126 DERIV=DERIV+DXFACT*(CAPQ3(2,I1,I2P)-CAPQ3(2,I1,I2M))
      GO TO 139
C
C
  131 DUM=DXFACT*L2
      GO TO (132,134,136),J
  132 DERIV=DERIV+DUM*(-3.*CAPQ1(2,I1,I2)+4.*CAPQ1(2,I1,I2+L2)-CAPQ1(2,I
     * 1,I2+2*L2))
      GO TO 139
  134 DERIV=DERIV+DUM*(-3.*CAPQ2(2,I1,I2)+4.*CAPQ2(2,I1,I2+L2)-CAPQ2(2,I
     * 1,I2+2*L2))
      GO TO 139
  136 DERIV=DERIV+DUM*(-3.*CAPQ3(2,I1,I2)+4.*CAPQ3(2,I1,I2+L2)-CAPQ3(2,I
     * 1,I2+2*L2))
  139 CONTINUE
C
C
      DERIV=DERIV+FORCEZ(J)
      ACC(J,I1,I2)=(DERIV+E1(J,I1,I2))*TERM
      D(J,I1,I2)=D(J,I1,I2)*TERMT+ACC(J,I1,I2)*TERM2
      Y(J,I1,I2)=Y(J,I1,I2)+D(J,I1,I2)
C                       VISCOUS DAMPING C1
      IF(TDAMP .EQ. 0.0)GOTO 298
      D(J,I1,I2)=D(J,I1,I2)-(D(J,I1,I2)+DJR)*C1
  298 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EQUIL2
C
C
C     EVALUATE MIDSURFACE GEOMETRIC QUANTITIES(BASE VECTORS, SURFACE
C     NORMAL, METRIC TENSOR, CURVATURE TENSOR, ETC.)
C
      IMPLICIT LOGICAL(Q)
C
      COMMON /CARTE/ YTEST,YNEW,YSAVE
      COMMON /CARTEL/ Y(3,20,20),D(3,20,20),Y2(3,20,20),D2(3,20,20)
      LEVEL 2,Y,D,Y2,D2
C
      COMMON /CTIME/ AUX(20),TIME,DELTAT,TIMEF,ITIME,ITIMEF,IAUX(20),
     * IOUT(20),QPRINT(20)
      COMMON /CTIMEL/ IPLAST(20,20),P(20,20),PPL(20,20)
      LEVEL 2,IPLAST,P,PPL
C
      COMMON/CTIMER/ITIMEC,ITIMER,DELTAP,DELX,OHR,UNH,HEE
     * ,TKEEP,HTHIK,QFINIS,QFINP,TSTART,YSTART,YDOTF
     * ,ES,RSTIV(4),NSTIV(4)
C
      COMMON /OPTERM/ MDISPL(4),MDISDR(4,3),PERIOD(4,3),DISPL(4,3),QDISP3
      COMMON /OPTERL/ PRSCRB(3,20,20)
      LEVEL 2,PRSCRB
C
      COMMON /INDEX/ NREAD,NWRITE,NPUNCH,NMESH1,NMESH2,N1,N2,NZ,N1M,N2M,
     * N1MM,N2MM,I1,I2,IZ,I1ZERO,I2ZERO,IRY1,IRY2,IRY3,IRY4,ISTR1,ISTR2,
     * ISTR3,ISTR4,IC1,IC2,ID1,ID2,IP1,IP2,IS1,IS2,K1,K2,K3,K4,KRUN,
     * KZSTOP,KYTEST,IDIR,I1TEST,I2TEST,KINITL
C
      COMMON /OPTION/ MAUXIL,MINGEO,MINVEL,MLOAD,MMATPR,MSPLOA,
     * MSPTEM,MTEMPE,MTHIKL,MIMPUL,ISTRES,INORML,ISTREZ
C
      COMMON /POLEGM/SNPOLE(4,3),VELPOL(4,3),NUMS(4),NPL,LOCPOL(4),QPOLEF
     * ,DYPOLE(4,3),YPOLE(4,3),D2POLE(4,3),Y2POLE(4,3),JPOLE(4)
     * ,QPOLET
C
      COMMON /PUSH/ FORCES(3),VELOC(3),RATIO,RATIOM,DX1,DX2,TEMP,DTEMP,
     * FSPACE,TSPACE,FINCND,FSTOP,TSTOP,THCOEF
      COMMON /PUSHL/ SQRAT(20,20),SQRAZ(20,20),FMAS11(20,20),
     * FMAS22(20,20),FMAS23(20,20),FMAS33(20,20)
      LEVEL 2,SQRAT,SQRAZ,FMAS11,FMAS22,FMAS23,FMAS33
C
      COMMON /QRCOND/ Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11,Q12,Q13,Q14,Q15
     * ,QFREE1,QFREE2,QFREE3,QFREE4,QFCORN,QQ1,QQ2                    ,Q
     * TRAC1,QTRAC2,QTRAC3,QTRAC4
C
      COMMON /SURNOM/ SNPR(3)
      COMMON /SURNOL/ SN(3,20,20)
      LEVEL 2,SN
C
      COMMON /S2/ STRESE(3)
      COMMON /S2L/ STRESL(3,20,20),STRESQ(2,20,20),STRESP(2,20,20)
      LEVEL 2,STRESL,STRESQ,STRESP
C
      COMMON /TENCOM/ YY(3,2),YYY(3,2,2),A(3,3),B(3,3),AA(3,3),BB(3,3),
     * BM(3,3),DA(3,3),DB(3,3),G(3,3),GG(3,3),DN(3),DD(3,2),DDD(3,2,2),
     * DGAM(3,3),DGAMMX(3,3),HN(3,3),HQ(2),TAU(3,3),TAUSBL(3,3)
     * ,DD2(3,2),DDD2(3,2,2),YY2(3,2),YYY2(3,2,2),YYU(3,2),DBM(3,3)
C
C
      COMMON /TNCOMP/ HM1(2,20,20),HM2(2,20,20),CAPQ1(2,20,20),
```

```
     * CAPQ2(2,20,20),CAPQ3(2,20,20),CAP2Q1(2,20,20),
     * CAP2Q2(2,20,20),CAP2Q3(2,20,20)
      LEVEL 2,HM1,HM2,CAPQ1,CAPQ2,CAPQ3,CAP2Q1,CAP2Q2,CAP2Q3
C
      COMMON /VELS/ Y3ACEL(3),Y3DOT(3),Y3ACPO(4,3)
      COMMON /VELSL/ VELO(3,20,20),ACC(3,20,20)
      LEVEL 2,VELO,ACC
C
      COMMON /ZACCEL/ SNDOT2(3,20,20),Y2DOT2(3,20,20)
      LEVEL 2,SNDOT2,Y2DOT2
C
      COMMON /DELC/ ICOUNT
      COMMON /DEL/ DELBAR(20,20)
      LEVEL 2,DELBAR
      COMMON /SURFOR/ E1(3,20,20),E2(3,20,20)
      LEVEL 2,E1,E2
      TERMT=DELTAT/DELTAP
      QQ1=I1.EQ.IRY1.OR.I1.EQ.IRY3
      QQ2=I2.EQ.IRY2.OR.I2.EQ.IRY4
      I1P=I1+1
      I2P=I2+1
      I1M=I1-1
      I2M=I2-1
      L1=ISIGN(1,N1-2*I1)
      L2=ISIGN(1,N2-2*I2)
      TERM2=DELTAT*DELTAT
      IF(ICOUNT .EQ. 0)GOTO 400
      CALL NACEL
      DO 298 J=1,3
C
C     NO Y2 STRAIN ON G9 SLIDING-CLAMPED EDGE IN MOISDR(K,J)=1 DIRECTION
      IF(QQ1.AND.I1.EQ.IRY1.AND.K1.EQ.9.AND.MOISDR(1,J).EQ.1) GO TO 390
      IF(QQ1.AND.I1.EQ.IRY3.AND.K3.EQ.9.AND.MOISDR(3,J).EQ.1) GO TO 390
      IF(QQ2.AND.I2.EQ.IRY2.AND.K2.EQ.9.AND.MOISDR(2,J).EQ.1) GO TO 390
      IF(QQ2.AND.I2.EQ.IRY4.AND.K4.EQ.9.AND.MOISDR(4,J).EQ.1) GO TO 390
C
C     TO AVOID ROUNDOFF IMPERFECTIONS
      IF (K1.EQ.7.AND.J.EQ.1) GO TO 390
      IF (K2.EQ.7.AND.J.EQ.2) GO TO 390
      IF(K1.EQ.5.AND.I1.EQ.2.AND.J.EQ.1) GO TO 390
      IF(K2.EQ.5.AND.I2.EQ.2.AND.J.EQ.2) GO TO 390
      IF(K3.EQ.5.AND.I1.EQ.N1M.AND.J.EQ.1)  GO TO 390
      IF(K4.EQ.5.AND.I2.EQ.N2M.AND.J.EQ.2)  GO TO 390
      IF (K1.EQ.6.AND.J.EQ.1) GO TO 390
      IF (K2.EQ.6.AND.J.EQ.2) GO TO 390
C
      DERIV=0.
C
      DX=DX1
      DXFACT=.5/DX
C
      IF(QQ1) GO TO 130
C
      GO TO (120,124,126), J
  120 DERIV=DERIV+DXFACT*(CAP2Q1(1,I1P,I2)-CAP2Q1(1,I1M,I2))
      GO TO 140
  124 DERIV=DERIV+DXFACT*(CAP2Q2(1,I1P,I2)-CAP2Q2(1,I1M,I2))
      GO TO 140
  126 DERIV=DERIV+DXFACT*(CAP2Q3(1,I1P,I2)-CAP2Q3(1,I1M,I2))
      GO TO 140
C
C
  130 DUM=DXFACT*L1
      GO TO (132,134,136), J
  132 DERIV=DERIV+DUM*(-3.*CAP2Q1(1,I1,I2)+4.*CAP2Q1(1,I1+L1,I2)-CAP2Q1(
     * 1,I1+2*L1,I2))
```

```
      GO TO 140
  134 DERIV=DERIV+DUM*(-3.*CAP2Q2(1,I1,I2)+4.*CAP2Q2(1,I1+L1,I2)-CAP2Q2(
     * 1,I1+2*L1,I2))
      GO TO 140
  136 DERIV=DERIV+DUM*(-3.*CAP2Q3(1,I1,I2)+4.*CAP2Q3(1,I1+L1,I2)-CAP2Q3(
     * 1,I1+2*L1,I2))
  140 CONTINUE
C
C
C
      DX=DX2
      DXFACT=.5/DX
C
      IF(QQ2) GO TO 151
C
      GO TO (142,144,146), J
  142 DERIV=DERIV+DXFACT*(CAP2Q1(2,I1,I2P)-CAP2Q1(2,I1,I2M))
      GO TO 160
  144 DERIV=DERIV+DXFACT*(CAP2Q2(2,I1,I2P)-CAP2Q2(2,I1,I2M))
      GO TO 160
  146 DERIV=DERIV+DXFACT*(CAP2Q3(2,I1,I2P)-CAP2Q3(2,I1,I2M))
      GO TO 160
C
C
  151 DUM=DXFACT*L2
      GO TO (152,154,156),J
  152 DERIV=DERIV+DUM*(-3.*CAP2Q1(2,I1,I2)+4.*CAP2Q1(2,I1,I2+L2)-CAP2Q1(
     * 2,I1,I2+2*L2))
      GO TO 160
  154 DERIV=DERIV+DUM*(-3.*CAP2Q2(2,I1,I2)+4.*CAP2Q2(2,I1,I2+L2)-CAP2Q2(
     * 2,I1,I2+2*L2))
      GO TO 160
  156 DERIV=DERIV+DUM*(-3.*CAP2Q3(2,I1,I2)+4.*CAP2Q3(2,I1,I2+L2)-CAP2Q3(
     * 2,I1,I2+2*L2))
  160 CONTINUE
C
      Y2DOT2(J,I1,I2)=(DERIV+E2(J,I1,I2)-STRESL(J,I1,I2)-FMAS23(I1,I2)
     * *Y3ACEL(J))/FMAS22(I1,I2)
C
      D2(J,I1,I2)=D2(J,I1,I2)*TERMT+TERM2*Y2DOT2(J,I1,I2)
      Y2(J,I1,I2)=Y2(J,I1,I2)+D2(J,I1,I2)
  390 CONTINUE
  298 CONTINUE
C     TAKE OUT NORMAL COMPONENT WHEN IT IS DESIRED TO BE ZERO
      IF(INORML.NE.1) GO TO 300
      DELNOR=SN(1,I1,I2)*D2(1,I1,I2)+SN(2,I1,I2)*D2(2,I1,I2)
     * +SN(3,I1,I2)*D2(3,I1,I2)
      DELSN1=DELNOR*SN(1,I1,I2)
      DELSN2=DELNOR*SN(2,I1,I2)
      DELSN3=DELNOR*SN(3,I1,I2)
      GOTO 410
  400 DELSN1=-DELBAR(I1,I2)*SN(1,I1,I2)
      DELSN2=-DELBAR(I1,I2)*SN(2,I1,I2)
      DELSN3=-DELBAR(I1,I2)*SN(3,I1,I2)
  410 D2(1,I1,I2)=D2(1,I1,I2)-DELSN1
      D2(2,I1,I2)=D2(2,I1,I2)-DELSN2
      D2(3,I1,I2)=D2(3,I1,I2)-DELSN3
      Y2DOT2(1,I1,I2)=Y2DOT2(1,I1,I2)-TERM2*DELSN1
      Y2DOT2(2,I1,I2)=Y2DOT2(2,I1,I2)-TERM2*DELSN2
      Y2DOT2(3,I1,I2)=Y2DOT2(3,I1,I2)-TERM2*DELSN3
      Y2(1,I1,I2)=Y2(1,I1,I2)-DELSN1
      Y2(2,I1,I2)=Y2(2,I1,I2)-DELSN2
      Y2(3,I1,I2)=Y2(3,I1,I2)-DELSN3
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRINT(INDEXX)
      IMPLICIT LOGICAL(Q)
C
      COMMON /ALLENE/ TOTAL,TOTKIN,TOTELA,TOTPLA,TOTWEX,TOTTEM,INFRGY
     * ,TOTVIS,TOTE1,TOTE2,ECHECK
     * ,DTM(3,3),SQRG,SQRA,SIGMSQ,AZ,BZ,CZ
C
      COMMON /CARTE/ YTEST,YNEW,YSAVE
      COMMON /CARTEL/ Y(3,20,20),D(3,20,20),Y2(3,20,20),D2(3,20,20)
      LEVEL 2,Y,D,Y2,D2
C
      COMMON /CTIME/ AUX(20),TIME,DELTAT,TIMEF,ITIME,ITIMEF,IAUX(20),
     * IOUT(20),QPRINT(20)
      COMMON /CTIMEL/ IPLAST(20,20),P(20,20),PPL(20,20)
      LEVEL 2,IPLAST,P,PPL
C
      COMMON /INDEX/ NREAD,NWRITE,NPUNCH,NMESH1,NMESH2,N1,N2,NZ,N1M,N2M,
     * N1MM,N2MM,I1,I2,IZ,I1ZERO,I2ZERO,IRY1,IRY2,IRY3,IRY4,ISTR1,ISTR2,
     * ISTR3,ISTR4,IC1,IC2,ID1,ID2,IP1,IP2,IS1,IS2,K1,K2,K3,K4,KRUN,
     * KZSTOP,KYTEST,IDIR,I1TEST,I2TEST,KINITL
C
      COMMON /PHYSCN/  EE,HNU,ALPHA,CONST,EXPON,FACTOR,RATE,RHO,
     * HLAMDA,COEFF(5),SIGMA(5),TM(3,3),TC(3,3),DELTA(3,3)
C
      COMMON/POLEGM/SNPOLE(4,3),VELPOL(4,3),NUMS(4),NPL,LOCPOL(4),QPOLE
     * ,DYPOLE(4,3),YPOLE(4,3),D2POLE(4,3),Y2POLE(4,3),JPOLE(4)
     * ,QPOLET
C
      COMMON /QBCOND/Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11,Q12,Q13,Q14,Q15
     * ,QFREE1,QFREE2,QFREE3,QFREE4,QFCORN,QQ1,QQ2                      ,Q
     * TRAC1,QTRAC2,QTRAC3,QTRAC4
C
      COMMON /QLOGIC/ QAUX(20),QZETA,QSTRES,QPLAST,QSENS1,QEQUIL,
     * QDIAGN,QINGEO,QINVEL,QLOAD,QMATPR,QTHIKL,QTEMPE,QSPTEM,QAUXI1,
     * QAUXI2,QSPLOA,QIMPUL,QSHARP,QPESO,QIRCH,QSHEAR
C
      COMMON /SURNOM/ SNPR(3)
      COMMON /SURNOL/ SN(3,20,20)
      LEVEL 2,SN
C
      COMMON /S2/ STRESE(3)
      COMMON /S2L/ STRESL(3,20,20),STRESQ(2,20,20),STRESP(2,20,20)
      LEVEL 2,STRESL,STRESQ,STRESP
C
      COMMON /TENCOM/ YY(3,2),YYY(3,2,2),A(3,3),B(3,3),AA(3,3),BB(3,3),
     * BM(3,3),DA(3,3),DB(3,3),G(3,3),GG(3,3),DN(3),DD(3,2),DDD(3,2,2),
     * DGAM(3,3),DGAMMX(3,3),HN(3,3),HQ(2),TAU(3,3),TAUSBL(3,3)
     * ,DD2(3,2),DDD2(3,2,2),YY2(3,2),YYY2(3,2,2),YYU(3,2),DBM(3,3)
C
      COMMON /THKNS/ Z,ZZ,ZCENTR,THICKN,ABSCIS(6,6),WEIGHT(6,6),
     * NGAUSS(4),IGAUSS,NLAYER,ILAYER,NSUBL(4),ISUBL
C
      COMMON /TNCOMP/ HM1(2,20,20),HM2(2,20,20),CAPQ1(2,20,20),
     * CAPQ2(2,20,20),CAPQ3(2,20,20),CAP2Q1(2,20,20),
     * CAP2Q2(2,20,20),CAP2Q3(2,20,20)
      LEVEL 2,HM1,HM2,CAPQ1,CAPQ2,CAPQ3,CAP2Q1,CAP2Q2,CAP2Q3
C
      COMMON /STPOUT/ ISB,L,LC,DISCR,TB(3,3),TN(3,3)
C
      COMMON /GPRINT/ GTYPE,GBASE(3,3),DGBASE(3,3)
C
      COMMON /THREE/ AGAM33
```

```
      COMMON /FRAC/ TAUF(3,3),TAUSPH,NUM,IPS1(10),IPS2(10)
C
      COMMON /MATRIX/ YLDFAC,ANUM
      COMMON /MATRIL/ LMAT(20,20,16),FMAT(6,20,20)
      LEVEL 2,LMAT,FMAT
C
      IF (ITIME .EQ. -1) RETURN
      II=INDEXX
      IF(INDEXX .GE. 4)II=4
      GOTO(1000,2000,3000,4000),II
C
C
C     POSITION AND TIME
C
 1000 III=ITIME/IOUT(1)*IOUT(1)-ITIME
      IF(III.NE.0.AND..NOT.QPRINT(1)) GO TO 701
      WRITE(NWRITE,1001)I1,I2,ITIME
 1001 FORMAT("1I1=",I5,"     I2=",I5,"     ITIME=",I5/)
  701 CONTINUE
C
C     FIRST AND SECOND  Y"S DERIVATIVES
C     FIRST AND SECOND DY"S DERIVATIVES
      III=ITIME/IOUT(2)*IOUT(2)-ITIME
      IF(III.NE.0.AND..NOT.QPRINT(2)) GO TO 702
      WRITE(NWRITE,100)
  100 FORMAT(//9X,"FIRST PARTIAL DERIVATIVE OF Y",20X,"SECOND PARTIAL DE
     *RIVATIVE OF Y"//12X,"YY(J,1)",8X,"YY(J,2)",19X,"YYY(J,1,1)",5X,
     *"YYY(J,1,2)",5X,"YYY(J,2,2)"/)
      WRITE(NWRITE,102) YY(1,1),YY(1,2),YYY(1,1,1),YYY(1,1,2),YYY(1,2,2)
      WRITE(NWRITE,102) YY(2,1),YY(2,2),YYY(2,1,1),YYY(2,1,2),YYY(2,2,2)
      WRITE(NWRITE,102) YY(3,1),YY(3,2),YYY(3,1,1),YYY(3,1,2),YYY(3,2,2)
  102 FORMAT(7X,2E15.6,12X,3E15.6)
      WRITE(NWRITE,103)
  103 FORMAT(//6X,"FIRST PARTIAL DERIVATIVE OF DELTA Y",14X,"SECOND PART
     *IAL DERIVATIVE OF DELTA Y"//12X,"DD(J,1)",8X,"DD(J,2)",19X,"DDD(J,
     *1,1)",5X,"DDD(J,1,2)",5X,"DDD(J,2,2)"/)
      WRITE(NWRITE,102) DD(1,1),DD(1,2),DDD(1,1,1),DDD(1,1,2),DDD(1,2,2)
      WRITE(NWRITE,102) DD(2,1),DD(2,2),DDD(2,1,1),DDD(2,1,2),DDD(2,2,2)
      WRITE(NWRITE,102) DD(3,1),DD(3,2),DDD(3,1,1),DDD(3,1,2),DDD(3,2,2)
      IF (QIRCH) GO TO 5101
      WRITE(NWRITE,5100)
 5100 FORMAT(//9X,"FIRST PARTIAL DERIVATIVE OF Y2",19X,"SECOND PARTIAL D
     *ERIVATIVE OF Y2",//12X,"YY2(J,1)",7X,"YY2(J,2)",18X,"YYY2(J,1,1)",
     * 4X,"YYY2(J,1,2)",4X,"YYY2(J,2,2)",/)
      WRITE(NWRITE,102) YY2(1,1),YY2(1,2),YYY2(1,1,1),YYY2(1,1,2),YYY2(1
     * ,2,2)
      WRITE(NWRITE,102) YY2(2,1),YY2(2,2),YYY2(2,1,1),YYY2(2,1,2),YYY2(2
     * ,2,2)
      WRITE(NWRITE,102) YY2(3,1),YY2(3,2),YYY2(3,1,1),YYY2(3,1,2),YYY2(3
     * ,2,2)
      WRITE(NWRITE,5103)
 5103 FORMAT(//5X,"FIRST PARTIAL DERIVATIVE OF DELTA Y2",14X,"SECOND PAR
     *TIAL DERIVATIVE OF DELTA Y2"//12X,"DD2(J,1)",7X,"DD2(J,2)",18X,"DD
     *D2(J,1,1)",4X,"DDD2(J,1,2)",4X,"DDD2(J,2,2)"/)
      WRITE(NWRITE,102) DD2(1,1),DD2(1,2),DDD2(1,1,1),DDD2(1,1,2),DDD2(1
     * ,2,2)
      WRITE(NWRITE,102) DD2(2,1),DD2(2,2),DDD2(2,1,1),DDD2(2,1,2),DDD2(2
     * ,2,2)
      WRITE(NWRITE,102) DD2(3,1),DD2(3,2),DDD2(3,1,1),DDD2(3,1,2),DDD2(3
     * ,2,2)
 5101 CONTINUE
  702 CONTINUE
C
C      FIRST AND SECOND METRIC TENSORS
C      FIRST AND SECOND METRIC TENSORS" INCREMENTS
C

      III=ITIME/IOUT(3)*IOUT(3)-ITIME
      IF(III.NE.0.AND..NOT.QPRINT(3)) GO TO 703
      WRITE(NWRITE,106)
  106 FORMAT(//14X,"FIRST METRIC TENSOR",29X,"SECOND METRIC TENSOR"/)
      WRITE(NWRITE,907) A(1,1),A(1,2),B(1,1),B(1,2)
  907 FORMAT(2X,"A(1,1)=",E13.6,"   A(1,2)=",E13.6,6X,"B(1,1)=",E13.6,
     *"   B(1,2)=",E13.6)
      WRITE(NWRITE,908) A(2,1),A(2,2),B(2,1),B(2,2)
  908 FORMAT(2X,"A(2,1)=",E13.6,"   A(2,2)=",E13.6,6X,"B(2,1)=",E13.6,
     *"   B(2,2)=",E13.6)
      WRITE(NWRITE,109)
  109 FORMAT(//9X,"FIRST METRIC TENSOR INCREMENT",19X,"SECOND METRIC TEN
     *SOR INCREMENT"/)
      WRITE(NWRITE,909) DA(1,1),DA(1,2),DB(1,1),DB(1,2)
  909 FORMAT(" DA(1,1)=",E13.6,"   DA(1,2)=",E13.6,"    DB(1,1)=",E13.6,
```

```
     *"   DB(1,2)=",E13.6)                                               PRINT144
      WRITE(NWRITE,910) DA(2,1),DA(2,2),DB(2,1),DB(2,2)                  PRINT145
  910 FORMAT(" DA(2,1)=",E13.6,"   DA(2,2)=",E13.6,"   DB(2,1)=",E13.6,PRINT146
     *"   DB(2,2)=",E13.6)                                               PRINT147
  703 CONTINUE                                                           PRINT148
C                                                                        PRINT149
C     STRESS RESULTANT                                                   PRINT150
C                                                                        PRINT151
      III=ITIME/IOUT(5)*IOUT(5)-ITIME                                    PRINT152
      IF(III.NE.0.AND..NOT.QPRINT(5)) GO TO 705                          PRINT153
      WRITE(NWRITE,118)                                                  PRINT154
  118 FORMAT(//" CONTRAVARIANT COMPONENTS OF THE RELATIVE STRESS-RESULTAPRINT155
     *NT TENSOR"/)                                                       PRINT156
      WRITE(NWRITE,917) HN(1,1),HN(1,2),HN(1,3)                          PRINT157
  917 FORMAT(1X,"HN(1,1)=",E13.6," HN(1,2)=",E13.6," HN(1,3)=",E13.6) PRINT158
      WRITE(NWRITE,918) HN(2,1),HN(2,2),HN(2,3)                          PRINT159
  918 FORMAT(1X,"HN(2,1)=",E13.6," HN(2,2)=",E13.6," HN(2,3)=",E13.6) PRINT160
  705 CONTINUE                                                           PRINT161
      RETURN                                                             PRINT162
C                                                                        PRINT163
C                                                                        PRINT164
C     RESULTANTS                                                         PRINT165
C                                                                        PRINT166
 2000 IF (ITIME.EQ.0) RETURN                                             PRINT167
      III=ITIME/IOUT(6)*IOUT(6)-ITIME                                    PRINT168
      IF(III.NE.0.AND..NOT.QPRINT(6)) GO TO 706                          PRINT169
      ITIMEP=ITIME-1                                                     PRINT170
      TIMEP=ITIMEP*DELTAT                                                PRINT171
C                                                                        PRINT172
C     WRITE TIME AND ITIME AT PRECFEDING STEP                            PRINT173
C                                                                        PRINT174
      WRITE(NWRITE,2) TIMEP,ITIMEP                                       PRINT175
      WRITE(NWRITE,920)                                                  PRINT176
  920 FORMAT(//36X,"TENSOR DEFINED IN          ",29X,"CONTRAVARIANT COMPPRINT177
     *ONENTS OF THE"/33X," THE PETROS  4  REPORT",36X,"RELATIVE MOMENT RPRINT178
     *ESULTANT TENSOR"//3X,"I1   I2",32X,"Q(LA,J,I1,I2)",45X,"H(LA,LB,I1PRINT179
     *,I2)"/)                                                            PRINT180
      DO 128 I1=1,N1                                                     PRINT181
      DO 128 I2=1,N2                                                     PRINT182
      WRITE(NWRITE,921) I1,I2,CAPQ1(1,I1,I2),CAPQ2(1,I1,I2),CAPQ3(1,I1, PRINT183
     * I2),HM1(1,I1,I2),HM1(2,I1,I2)                                     PRINT184
  921 FORMAT(2I5,"   CAPQ(1,1)=",E13.6,"  CAPQ(1,2)=",E13.6,             PRINT185
     *"  CAPQ(1,3)=",E13.6,"  HM11=",E13.6,"  HM12=",E13.6)              PRINT186
      WRITE(NWRITE,922) CAPQ1(2,I1,I2),CAPQ2(2,I1,I2),CAPQ3(2,I1,I2),    PRINT187
     * HM2(1,I1,I2),HM2(2,I1,I2)                                         PRINT188
  922 FORMAT(13X,"CAPQ(2,1)=",E13.6,"  CAPQ(2,2)=",E13.6,"  CAPQ(2,3)=" PRINT189
     *,E13.6,"  HM21=",E13.6,"  HM22=",E13.6)                            PRINT190
      IF (QIRCH) GO TO 128                                               PRINT191
      WRITE(NWRITE,821) I1,I2,CAP2Q1(1,I1,I2),CAP2Q2(1,I1,I2),CAP2Q3(1,IPRINT192
     * 1,I2),STRESP(1,I1,I2),STRESP(2,I1,I2)                             PRINT193
  821 FORMAT(2I5,"  CAP2Q(1,1)=",E13.6," CAP2Q(1,2)=",E13.6," CAP2Q(1,3)PRINT194
     *=",E13.6,"  P(1)=",E13.6,"  P(2)=",E13.6)                          PRINT195


      WRITE(NWRITE,822)CAP2Q1(2,I1,I2),CAP2Q2(2,I1,I2),CAP2Q3(2,I1,I2), PRINT196
     * STRESQ(1,I1,I2),STRESQ(2,I1,I2)                                   PRINT197
  822 FORMAT(12X,"CAP2Q(2,1)=",E13.6," CAP2Q(2,2)=",E13.6,              PRINT198
     *" CAP2Q(2,3)=",E13.6,"  Q(1)=",E13.6,"  Q(2)=",E13.6)              PRINT199
      WRITE(NWRITE,823) I1,I2,STRESL(1,I1,I2),STRESL(2,I1,I2),STRESL(3, PRINT200
     * I1,I2)                                                            PRINT201
  823 FORMAT(2I5,"   STRESL(1)=",E13.6,"  STRESL(2)=",E13.6,"  STRESL(3)PRINT202
     *=",E13.6)                                                          PRINT203
  128 CONTINUE                                                           PRINT204
  706 CONTINUE                                                           PRINT205
C                                                                        PRINT206
C     NORMAL                                                             PRINT207
C                                                                        PRINT208
      III=ITIME/IOUT(7)*IOUT(7)-ITIME                                    PRINT209
      IF(III.NE.0.AND..NOT.QPRINT(7)) GO TO 707                          PRINT210
      ITIMEP=ITIME-1                                                     PRINT211
      TIMEP=ITIMEP*DELTAT                                                PRINT212
C                                                                        PRINT213
C     WRITE TIME AND ITIME AT PRECFEDING STEP                            PRINT214
C                                                                        PRINT215
      WRITE(NWRITE,2) TIMEP,ITIMEP                                       PRINT216
      WRITE(NWRITE,1002)                                                 PRINT217
 1002 FORMAT(//17X,"COMPONENTS OF THE SURFACE NORMAL"//3X,"I1   I2",    PRINT218
     *6X,"SN(1)",10X,"SN(2)",10X,"SN(3)"/)                               PRINT219
      DO 1007 I1=1,N1                                                    PRINT220
      DO 1007 I2=1,N2                                                    PRINT221
```

```
      WRITE(NWRITE,1003) I1,I2, SN(1,I1,I2),SN(2,I1,I2),SN(3,I1,I2)
 1003 FORMAT(2I5,3E15.6)
 1007 CONTINUE
      IF(.NOT.QR) GO TO 707
      DO 1010 I=1,NPL
      K=LOCPOL(I)
 1010 WRITE(NWRITE,1020) K, SNPOLE(K,1),SNPOLE(K,2),SNPOLE(K,3)
 1020 FORMAT(" POLE ON"/"  SIDE",I2," =",3E15.6)
  707 CONTINUE
C
C     POSITIONS AND INCREMENTS
C
      III=ITIME/IOUT(8)*IOUT(8)-ITIME
      IF(III.NE.0.AND..NOT.QPRINT(8)) GO TO 708
      WRITE(NWRITE,2) TIME,ITIME
    2 FORMAT("1",8X,"TIME=",E13.6,8X" ITIME=",I5/)
      WRITE(NWRITE,3)
    3 FORMAT(/48X,'CARTESIAN COORDINATES'//30X,'POSITION',37X,'CHANGE IN
     * POSITION'//3X,'I1   I2',8X,'Y1',13X,'Y2',13X,'Y3',18X,'D1',13X,'D
     *2',13X,'D3',12X,'P',10X,'IPLAST*'/)
      WRITE(NWRITE,4)((I1,I2,Y(1,I1,I2),Y(2,I1,I2),Y(3,I1,I2),D(1,I1,I2)
     * ,D(2,I1,I2),D(3,I1,I2),P(I1,I2),IPLAST(I1,I2),
     * I2=IRY2,IRY4),I1=IRY1,IRY3)
    4 FORMAT(2I5,3E15.6,'     ',4E15.6,3X,I5)
      IF (QIRCH) GO TO 1004
      WRITE(NWRITE,1053)
 1053 FORMAT(////52X,"Y2 COORDINATES   "//30X,"POSITION",37X,"CHANGE IN
     *POSITION"// 3X,"I1   I2",6X,"Y2(1)",10X,"Y2(2)",10X,"Y2(3)",15X,
     *"D2(1)",10X,"D2(2)",10X,"D2(3)          IPLAST *"/)
      WRITE(NWRITE,5)((I1,I2,Y2(1,I1,I2),Y2(2,I1,I2),Y2(3,I1,I2),D2(1,I1
     * ,I2),D2(2,I1,I2),D2(3,I1,I2),IPLAST(I1,I2),I2=IRY2,IRY4),I1=
     * IRY1,IRY3)
    5 FORMAT(2I5,3E15.6,'     ',3E15.6,3X,I5)
 1004 CONTINUE
      IF(NPL.EQ.0) GO TO 929
      DO 927 I=1,NPL
      K=LOCPOL(I)
      WRITE(NWRITE,928)K,YPOLE(K,1),YPOLE(K,2),YPOLE(K,3),DYPOLE(K,1),
     * DYPOLE(K,2),DYPOLE(K,3)
  928 FORMAT("0POLE ON SIDE",I2," HAS POSITION AND INCREMENTS ="/
     * 10X,3E15.6,"     ",3E15.6)
      IF (QIRCH) GO TO 927
      WRITE(NWRITE,1928)K,Y2POLE(K,1),Y2POLE(K,2),Y2POLE(K,3),D2POLE(K,1
     * ),D2POLE(K,2),D2POLE(K,3)
 1928 FORMAT(" POLE ON SIDE",I2,", Y2 POSITION AND INCREMENTS ="/
     * 10X,3E15.6,"     ",3E15.6)
  927 CONTINUE
  929 WRITE(NWRITE,930)
  930 FORMAT(/22X,"* IPLAST GREATER THAN ZERO INDICATES PLASTICITY AT TH
     *IS TIME STEP")
  708 CONTINUE
C
C             WRITE LMATRIX
      III=ITIME/IOUT(14)*IOUT(14)-ITIME
      IF(III.NE.0 .AND. .NOT.QPRINT(14))GOTO 730
      WRITE(NWRITE,711) TIME,ITIME
      IZ=0
      DO 710 ILAYER=1,NLAYER
      NGAUSL=NGAUSS(ILAYER)
      DO 710 IGAUSS=1,NGAUSL
      NSBL=NSUBL(ILAYER)
      DO 710 ISB=1,NSBL
      IZ=IZ+1
      WRITE(NWRITE,715) ILAYER,ILAYER,IGAUSS,ISB,(I2,I2=IRY2,IRY4)
      DO 710 I1=IRY1,IRY3
      WRITE(NWRITE,720) I1,(LMAT(I1,I2,IZ),I2=IRY2,IRY4)
  710 CONTINUE
  711 FORMAT('1',8X,'TIME=',E13.6,8X,' ITIME=',I5/
     *8X,'SUBDIVISIONS OF TIME INCREMENT IN STRESS'/)
  715 FORMAT(//20X,'LMAT(I1,I2,',I2,')  LAYER',I2,' GAUSS PT.',I2,
     *' SUBLAYER',I2//'   I1   I2=',40I3/)
  720 FORMAT(' ',I4,5X,40I3)
C              WRITE FMAT
  730 III=ITIME/IOUT(15)*IOUT(15)-ITIME
      IF(III.NE.0 .AND. .NOT.QPRINT(15))GOTO 760
      WRITE(NWRITE,740) TIME,ITIME
      DO 735 ILAYER=1,NLAYER
      NGAUSL=NGAUSS(ILAYER)
      DO 735 IGAUSS=1,NGAUSL
```

```
      WRITE(NWRITE,745) ILAYER,IGAUSS,(I2,I2=ISTR2,ISTR4)
      DO 735 I1=ISTR1,ISTR3
      WRITE(NWRITE,750) I1,(FMAT(IGAUSS,I1,I2),I2=ISTR2,ISTR4)
  735 CONTINUE
  740 FORMAT('1',8X,'TIME=',E13.6,8X,' ITIME=',I5/)
  745 FORMAT(//20X,'FMAT(I1,I2,',I2,')  GAUSS PT.',I2,
     *//'     I1 I2=',40I3)
  750 FORMAT('  ',I4,6X,40A3)
  760 CONTINUE
      RETURN
C
C     STRESSES
C
 3000 IF (ITIME.EQ.0) RETURN
      III=ITIME/IOUT(4)*IOUT(4)-ITIME
      IF(III.NE.0.AND..NOT.QPRINT(4)) GO TO 3003
      IF(ILAYER.EQ.1.AND.IGAUSS.EQ.1 .AND. ISUBL.EQ.1) WRITE(NWRITE,932)
  932 FORMAT("1",14X,"CONTRAVARIANT COMPONENTS OF THE STRESS TENSOR"// "
     * LAYER  GAUSS STATION  SUBLAYER    TAUSBL(I,1)    TAUSBL(I,2)    T
     *AUSBL(I,3)"/)
      WRITE(NWRITE,1933) ILAYER,IGAUSS,ISUBL
 1933 FORMAT(2X,I2,8X,I2,11X,I2)
      WRITE(NWRITE,1934) ((TAUSBL(I,J),J=1,3),I=1,3)
 1934 FORMAT(34X,3E15.6)
      NSBL=NSUBL(ILAYER)
      IF(ISUBL.EQ.NSBL) WRITE(NWRITE,3002) ILAYER,IGAUSS,
     * ((TAU(I,J),J=1,3),I=1,3)
 3002 FORMAT(/" CONTRAVARIANT COMP. OF THE STRESS TENSOR, TAU(I,J), FOR
     *LAYER",I2,", GAUSS STAT.",I2," ="/(85X,3E15.6))
 3003 RETURN
C
 4000 IF(ITIME .EQ. 0)RETURN
      IF(INDEXX .EQ. 11)GOTO 8300
      III=ITIME/IOUT(12)*IOUT(12)-ITIME
      IF(III.NE.0 .AND. .NOT.QPRINT(12))GOTO 4010
      II=(INDEXX-3)
      GOTO(4100,5000,6000,7000,8000,8100,8200),II
C
 4100 WRITE(NWRITE,4001) ISB
 4001 FORMAT(' SUBROUTINE STRESS'/' ISB=',I3,/)
      WRITE(NWRITE,4002) ((TN(J,I),I=1,3),J=1,3)
 4002 FORMAT(' TN(1,1)=',E22.15,' TN(1,2)=',E22.15,' TN(1,3)=',E22.15/
     *       ' TN(2,1)=',E22.15,' TN(2,2)=',E22.15,' TN(2,3)=',E22.15/
     *       ' TN(3,1)=',E22.15,' TN(3,2)=',E22.15,' TN(3,3)=',E22.15)
 4010 RETURN
C
 5000 WRITE(NWRITE,5005) ((TR(J,I),I=1,3),J=1,3)
 5005 FORMAT(' TR(1,1)=',E22.15,' TR(1,2)=',E22.15,' TR(1,3)=',E22.15/
     *       ' TR(2,1)=',E22.15,' TR(2,2)=',E22.15,' TR(2,3)=',E22.15/
     *       ' TR(3,1)=',E22.15,' TR(3,2)=',E22.15,' TR(3,3)=',E22.15)
      WRITE(NWRITE,5006) ISB,L,LC,CZ,SIGMSQ
 5006 FORMAT('   *** ISB=',I3,' L=',I3,' LC=',I3,' CZ=',E22.15,' SIGMSQ=
     *',E22.15)
      RETURN
C
 6000 WRITE(NWRITE,6005)AZ,BZ,DISCR
 6005 FORMAT(' AZ=',E22.15,' BZ=',E22.15,' DISCR=',E22.15)
      WRITE(NWRITE,6006) ((TC(J,I),I=1,3),J=1,3)
 6006 FORMAT(' TC(1,1)=',E22.15,' TC(1,2)=',E22.15,' TC(1,3)=',E22.15/
     *       ' TC(2,1)=',E22.15,' TC(2,2)=',E22.15,' TC(2,3)=',E22.15/
     *       ' TC(3,1)=',E22.15,' TC(3,2)=',E22.15,' TC(3,3)=',E22.15)
      RETURN
C
 7000 WRITE(NWRITE,7001) HLAMDA
 7001 FORMAT(' HLAMDA=',E22.15)
      WRITE(NWRITE,7006) ((TM(J,I),I=1,3),J=1,3)
 7006 FORMAT(' TM(1,1)=',E22.15,' TM(1,2)=',E22.15,' TM(1,3)=',E22.15/
     *       ' TM(2,1)=',E22.15,' TM(2,2)=',E22.15,' TM(2,3)=',E22.15/
     *       ' TM(3,1)=',E22.15,' TM(3,2)=',E22.15,' TM(3,3)=',E22.15)
      RETURN
C
 8000 WRITE(NWRITE,8001) ITIME,TIME,IGAUSS,I1,I2
 8001 FORMAT(' ITIME=',I4,' TIME=',E15.8,' IGAUSS=',I4,' I1=',I4,' I2=',
     *I4)
      WRITE(NWRITE,8002)
 8002 FORMAT(' SUBROUTINE ZETA')
      WRITE(NWRITE,8003) ((GBASE(J,I),I=1,3),J=1,3)
 8003 FORMAT(' GBASE(1,1)=',E22.15,' GBASE(1,2)=',E22.15,' GBASE(1,3)=',
```

```
     *E22.15/                                                          PRINT379
     *' GBASE(2,1)=',E22.15,' GBASE(2,2)=',E22.15,' GBASE(2,3)=',E22.15/PRINT380
     *' GBASE(3,1)=',E22.15,' GBASE(3,2)=',E22.15,' GBASE(3,3)=',E22.15)PRINT381
      WRITE(NWRITE,8004) ((DGBASE(J,I),I=1,3),J=1,3)                   PRINT382
 8004 FORMAT(' DGBASE(1,1)=',E22.15,' DGBASE(1,2)=',E22.15,' DGBASE(1,3)PRINT383
     *=',E22.15/                                                       PRINT384
     *' DGBASE(2,1)=',E22.15,' DGBASE(2,2)=',E22.15,' DGBASE(2,3)=',E22.PRINT385
     *15/                                                              PRINT386
     *' DGBASE(3,1)=',E22.15,' DGBASE(3,2)=',E22.15,' DGBASE(3,3)=',E22.PRINT387
     *15)                                                              PRINT388
      WRITE(NWRITE,8005) ((G(J,I),I=1,3),J=1,3)                        PRINT389
 8005 FORMAT(' G(1,1)=',E22.15,' G(1,2)=',E22.15,' G(1,3)=',E22.15/    PRINT390
     *       ' G(2,1)=',E22.15,' G(2,2)=',E22.15,' G(2,3)=',E22.15/    PRINT391
     *       ' G(3,1)=',E22.15,' G(3,2)=',E22.15,' G(3,3)=',E22.15)    PRINT392
      WRITE(NWRITE,8006) GTYPE                                         PRINT393
 8006 FORMAT(' GTYPE=',E22.15)                                         PRINT394
      WRITE(NWRITE,8007) ((GG(J,I),I=1,3),J=1,3)                       PRINT395
 8007 FORMAT(' GG(1,1)=',E22.15,' GG(1,2)=',E22.15,' GG(1,3)=',E22.15/ PRINT396
     *       ' GG(2,1)=',E22.15,' GG(2,2)=',E22.15,' GG(2,3)=',E22.15/ PRINT397
     *       ' GG(3,1)=',E22.15,' GG(3,2)=',E22.15,' GG(3,3)=',E22.15) PRINT398
      WRITE(NWRITE,8008) ((DGAM(J,I),I=1,3),J=1,3)                     PRINT399
 8008 FORMAT(' DGAM(1,1)=',E22.15,' DGAM(1,2)=',E22.15,' DGAM(1,3)=',E22PRINT400
     *.15/                                                             PRINT401
     *' DGAM(2,1)=',E22.15,' DGAM(2,2)=',E22.15,' DGAM(2,3)=',E22.15/  PRINT402
     *' DGAM(3,1)=',E22.15,' DGAM(3,2)=',E22.15,' DGAM(3,3)=',E22.15)  PRINT403
      WRITE(NWRITE,8009) ((DGAMMX(J,I),I=1,3),J=1,3)                   PRINT404
 8009 FORMAT(' DGAMMX(1,1)=',E22.15,' DGAMMX(1,2)=',E22.15,' DGAMMX(1,3)PRINT405
     *=',E22.15/                                                       PRINT406
     *' DGAMMX(2,1)=',E22.15,' DGAMMX(2,2)=',E22.15,' DGAMMX(2,3)=',E22.PRINT407
     *15/                                                              PRINT408
     *' DGAMMX(3,1)=',E22.15,' DGAMMX(3,2)=',E22.15,' DGAMMX(3,3)=',E22.PRINT409
     *15)                                                              PRINT410
      RETURN                                                           PRINT411
C                                                                      PRINT412
 8100 WRITE(NWRITE,8101) DGAMMX(3,3)                                   PRINT413
 8101 FORMAT(' (REVISED) DGAMMX(3,3)=',E22.15)                         PRINT414
      RETURN                                                           PRINT415
C                                                                      PRINT416
 8200 WRITE(NWRITE,8201) AGAM33                                        PRINT417
 8201 FORMAT(' AGAM33=',E22.15)                                        PRINT418
      RETURN                                                           PRINT419
C                                                                      PRINT420
C               MIXED TENSOR STRESSES                                  PRINT421
C                                                                      PRINT422
 8300 WRITE(NWRITE,8320) TIME,ITIME,I1,I2,IGAUSS,TAUSPH                PRINT423
      WRITE(NWRITE,8325) ((TAUF(K,J),J=1,3),K=1,3)                     PRINT424
      RETURN                                                           PRINT425
C                                                                      PRINT426
 8320 FORMAT(' TIME=',E15.6,3X,'ITIME =',I5,' I1=',I3,' I2=',I3,       PRINT427
     *' GAUSS PT. =',I3,'  TAUSPH =',F10.1)                            PRINT428
 8325 FORMAT(55X,'MIXED TENSOR STRESSES'/                              PRINT429
     *39X,' TAUF(1,1)=',F10.1,' TAUF(1,2)=',F10.1,' TAUF(1,3)=',F10.1/ PRINT430
     *39X,' TAUF(2,1)=',F10.1,' TAUF(2,2)=',F10.1,' TAUF(2,3)=',F10.1/ PRINT431
     *39X,' TAUF(3,1)=',F10.1,' TAUF(3,2)=',F10.1,' TAUF(3,3)=',F10.1) PRINT432
      END                                                              PRINT433
```

```
      SUBROUTINE MAXMIN(TAU,IGO)
      IMPLICIT LOGICAL (Q)
C
      COMMON /INDEX/ NREAD,NWRITE,NPUNCH,NMESH1,NMESH2,N1,N2,N2,N1M,N2M,
     * N1MM,N2MM,I1,I2,IZ,I1ZERO,I2ZERO,IRY1,IRY2,IRY3,IRY4,ISTR1,ISTR2,
     * ISTR3,ISTR4,IC1,IC2,IO1,IO2,IP1,IP2,IS1,IS2,K1,K2,K3,K4,KRUN,
     * KZSTOP,KYTEST,IDIM,I1TEST,I2TEST,KINITL
C
      COMMON /TRANS/ Z,ZZ,ZCENTR,THICKN,ABSCIS(6,6),WEIGHT(6,6),
     * NGAUSS(4),IGAUSS,NLAYER,ILAYER,NSUBL(4),ISUBL
C
      COMMON /CTIME/ AUX(20),TIME,DELTAT,TIMEF,ITIME,ITIMEF,IAUX(20),
     * IOUT(20),QPRINT(20)
      COMMON /CTIMEL/ IPLAST(20,20),P(20,20),PPL(20,20)
      LEVEL 2,IPLAST,P,PPL
C
      DIMENSION TAU(3,3),TAUMAX(3,3),TAUMIN(3,3),I1MAX(3,3),I1MIN(3,3),
     * I2MAX(3,3),I2MIN(3,3),IGMAX(3,3),IGMIN(3,3),ITIM(3,3),TIM(3,3)
     * ,ITIMM(3,3),TIMM(3,3)
C
      DATA QM/.FALSE./,MAX/' MAX'/,MIN/' MIN'/
C
      GOTO(5,35),IGO
C
    5 IF(QM)GOTO 20
      DO 10 I=1,3
      DO 10 J=1,3
      TAUMAX(I,J)=0.0
      TAUMIN(I,J)=0.0
      I1MAX(I,J)=0
      I1MIN(I,J)=0
      I2MAX(I,J)=0
      I2MIN(I,J)=0
      IGMIN(I,J)=0
      IGMAX(I,J)=0
   10 CONTINUE
      QM=.TRUE.
C
   20 DO 30 I=1,3
      DO 30 J=1,3
      Q11=TAU(I,J) .GT. TAUMAX(I,J)
      IF(.NOT. Q11)GOTO 25
      TAUMAX(I,J)=TAU(I,J)
      I1MAX(I,J)=I1
      I2MAX(I,J)=I2
      ITIM(I,J)=ITIME
      TIM(I,J)=TIME
      IGMAX(I,J)=IGAUSS
   25 Q11=TAU(I,J) .LT. TAUMIN(I,J)
      IF(.NOT. Q11)GOTO 30
      TAUMIN(I,J)=TAU(I,J)
      I1MIN(I,J)=I1
      I2MIN(I,J)=I2
      ITIMM(I,J)=ITIME
      TIMM(I,J)=TIME
      IGMIN(I,J)=IGAUSS
   30 CONTINUE
      RETURN
C
   35 WRITE(NWRITE,100) MAX
   40 DO 45 I=1,3
      DO 45 J=1,3
      WRITE(NWRITE,110) TIM(I,J),ITIM(I,J),I1MAX(I,J),I2MAX(I,J)
     * ,IGMAX(I,J),I,J,TAUMAX(I,J)
   45 CONTINUE
C
      WRITE(NWRITE,100) MIN
      DO 50 I=1,3
      DO 50 J=1,3
      WRITE(NWRITE,110) TIMM(I,J),ITIMM(I,J),I1MIN(I,J),I2MIN(I,J)
     * ,IGMIN(I,J),I,J,TAUMIN(I,J)
   50 CONTINUE
      RETURN
C
  100 FORMAT(/'     ',A4,'IMUM MIXED TENSOR STRESSES'/)
  110 FORMAT('    TIME=',E13.6,' ITIME=',I5,' I1=',I2,' I2=',I2,
     *' GAUSS PT.',I2,' TAUF(',I1,',',I1,')=',E14.3)
      END
```

```
      SUBROUTINE SFORCE (CS,THICKN,YY,YY2,I1,I2,SURFGG)
C         SURFACE FORCES (ONLY FOR SHEAR OPTION)
      COMMON /CARTE/ YTEST,YNEW,YSAVE
      COMMON /CARTEL/ Y(3,20,20),D(3,20,20),Y2(3,20,20),D2(3,20,20)
      LEVEL 2,Y,D,Y2,D2
C
      COMMON /PUSH/ FORCES(3),VELOC(3),RATIO,RATIOM,DX1,DX2,TEMP,DTEMP,
     * FSPACE,TSPACE,FINCNO,FSTOP,TSTOP,THCOEF
      COMMON /PUSHL/ SQRAT(20,20),SQRAZ(20,20),FMAS11(20,20),
     * FMAS22(20,20),FMAS23(20,20),FMAS33(20,20)
      LEVEL 2,SQRAT,SQRAZ,FMAS11,FMAS22,FMAS23,FMAS33
C
      COMMON /SURNOM/ SNPR(3)
      COMMON /SURNOL/ SN(3,20,20)
      LEVEL 2,SN
C
      COMMON /SURFOR/ E1(3,20,20),E2(3,20,20)
      LEVEL 2,E1,E2
C
      DIMENSION DZ(3),CS(3,2),GBASE(3,3),G(3,3),GG(3,3),YY(3,2),YY2(3,2)
     * ,SUME(3)
C
      DO 3 J=1,3
      E1(J,I1,I2)=0.0
      E2(J,I1,I2)=0.0
    3 CONTINUE
C
      DO 50 IZZ=1,2
      ZZ=0
      CALL ERASE (DZ,3)
      DZ(3)=1.0
      GOTO (25,30),IZZ
   25 ZZ=-.5*THICKN
      SIGN=1.0
      CALL LOADL
      GOTO 34
   30 ZZ=.5*THICKN
      SIGN=-1.0
      CALL LOAD
   34 DO 35 J=1,3
      GBASE(1,J)=YY(J,1)-ZZ*CS(J,1)
      GBASE(2,J)=YY(J,2)-ZZ*CS(J,2)
      GBASE(3,J)=SN(J,I1,I2)
C
      GBASE(1,J)=GBASE(1,J)+DZ(1)*Y2(J,I1,I2)+ZZ*YY2(J,1)
      GBASE(2,J)=GBASE(2,J)+DZ(2)*Y2(J,I1,I2)+ZZ*YY2(J,2)
      GBASE(3,J)=GBASE(3,J)+DZ(3)*Y2(J,I1,I2)
   35 CONTINUE
      G(1,1)=GBASE(1,1)*GBASE(1,1)+GBASE(1,2)*GBASE(1,2)
     * +GBASE(1,3)*GBASE(1,3)
      G(1,2)=GBASE(1,1)*GBASE(2,1)+GBASE(1,2)*GBASE(2,2)
     * +GBASE(1,3)*GBASE(2,3)
      G(1,3)=GBASE(1,1)*GBASE(3,1)+GBASE(1,2)*GBASE(3,2)
     * +GBASE(1,3)*GBASE(3,3)
      G(2,1)=G(1,2)
      G(2,2)=GBASE(2,1)*GBASE(2,1)+GBASE(2,2)*GBASE(2,2)
     * +GBASE(2,3)*GBASE(2,3)
      G(2,3)=GBASE(2,1)*GBASE(3,1)+GBASE(2,2)*GBASE(3,2)
     * +GBASE(2,3)*GBASE(3,3)
      G(3,1)=G(1,3)
      G(3,2)=G(2,3)
      G(3,3)=GBASE(3,1)*GBASE(3,1)+GBASE(3,2)*GBASE(3,2)
```

```
     * +GBASE(3,3)*GBASE(3,3)                                          SFORCE63
C     G=GTYPE                                                          SFORCE64
      GTYPE=G(1,1)*(G(2,2)*G(3,3)-G(2,3)**2)+G(1,2)*(G(1,3)*G(2,3)-    SFORCE65
     * G(1,2)*G(3,3))+G(1,3)*(G(1,2)*G(2,3)-G(1,3)*G(2,2))             SFORCE66
      SQRG=SQRT(GTYPE)                                                 SFORCE67
      GTYPE=1./GTYPE                                                   SFORCE68
      GG(1,1)=(G(2,2)*G(3,3)-G(2,3)**2)*GTYPE                          SFORCE69
      GG(1,2)=(G(1,3)*G(2,3)-G(1,2)*G(3,3))*GTYPE                      SFORCE70
      GG(2,1)=GG(1,2)                                                  SFORCE71
      GG(1,3)=(G(1,2)*G(2,3)-G(1,3)*G(2,2))*GTYPE                      SFORCE72
      GG(3,1)=GG(1,3)                                                  SFORCE73
      GG(2,2)=(G(1,1)*G(3,3)-G(1,3)**2)*GTYPE                          SFORCE74
      GG(2,3)=(G(1,2)*G(1,3)-G(1,1)*G(3,2))*GTYPE                      SFORCE75
      GG(3,2)=GG(2,3)                                                  SFORCE76
      GG(3,3)=(G(1,1)*G(2,2)-G(1,2)**2)*GTYPE                          SFORCE77
      IF(IZZ .EQ. 2) SURFGG=GG(3,3)                                    SFORCE78
      CALL ERASE (SUME,3)                                              SFORCE79
      DO 40 K=1,3                                                      SFORCE80
      DO 40 I=1,3                                                      SFORCE81
      SUME(K)=SUME(K)+GG(3,I)*GBASE(I,K)                               SFORCE82
   40 CONTINUE                                                         SFORCE83
      DO 45 J=1,3                                                      SFORCE84
      IF(SN(J,I1,I2) .EQ. 0.0)GOTO 45                                  SFORCE85
      PRESS=-(FORCES(J)/SN(J,I1,I2))                                   SFORCE86
      TEMP=SQRG*SUME(J)*PRESS                                          SFORCE87
      E1(J,I1,I2)=E1(J,I1,I2)+SIGN*TEMP                                SFORCE88
      E2(J,I1,I2)=E2(J,I1,I2)+SIGN*ZZ*TEMP                             SFORCE89
   45 CONTINUE                                                         SFORCE90
   50 CONTINUE                                                         SFORCE91
      RETURN                                                           SFORCE92
      END                                                              SFORCE93
```

```
      SUBROUTINE STRESS (JD1,JD2,JD3,TAU11,TAU12,TAU13,
     * TAU21,TAU22,TAU23,TAU31,TAU32,TAU33)
C
C     EVALUATE STRESS INCREMENTS AND STRESSES
C
      IMPLICIT LOGICAL(Q)
C
      COMMON/ ALLENE /TOTAL,TOTKIN,TOTELA,TOTPLA,TOTWEX,TOTTEM,INERGY
     * , TOTVIS,TOTE1,TOTE2 ,ECHECK
     * ,DTM(3,3),SQRG,SQRA,SIGMSQ,AZ,BZ,CZ
C
      COMMON /CTIME/ AUX(20),TIME,DELTAT,TIMEF,ITIME,ITIMEF,IAUX(20),
     * IOUT(20),QPRINT(20)
      COMMON /CTIMEL/ IPLAST(20,20),P(20,20),PPL(20,20)
      LEVEL 2,IPLAST,P,PPL
C
      COMMON/CTIMER/ITIMEC,ITIMER,DELTAP,DELX,OHR,UNH,HEE
     * ,TKEEP,MTHIK,QFINIS,QFINP,TSTART,YSTART,YOUTF
     * ,ES,BSTIV(4),NSTIV(4)
C
      COMMON /INDEX/ NREAD,NWRITE,NPUNCH,NMESH1,NMESH2,N1,N2,NZ,N1M,N2M,
     * N1MM,N2MM,I1,I2,IZ,I1ZERO,I2ZERO,IRY1,IRY2,IRY3,IRY4,ISTR1,ISTR2,
     * ISTR3,ISTR4,IC1,IC2,ID1,ID2,IP1,IP2,IS1,IS2,K1,K2,K3,K4,KRUN,
     * KZSTOP,KYTEST,IDIR,I1TEST,I2TEST,KINITL
C
      COMMON /OPTION/ MAUXIL,MINGEO,MINVEL,MLOAD,MMATPR,MSPLOA,
     * MSPTEM,MTEMPE,MTHIKL,MIMPUL,ISTRES,INCRPL,ISTREZ
      COMMON /OPTFRA/ IFRACT,QFRACT
      COMMON /MAII/ NCONT,NRITE,NTRAIN,MRITE,NDELP,ETAD1,ETAD2,NSTRN
     * ,FACTOM,MQ,FACTON,NQ,FKIN
C
      COMMON /PHYSCN/ EE,HNU,ALPHA,CONST,EXPON,FACTOR,RATE,RHO,
     * HLAMDA,COEFF(5),SIGMA(5),TM(3,3),TC(3,3),DELTA(3,3)
C
      COMMON /PUSH/ FORCES(3),VELOC(3),RATIO,RATIOM,DX1,DX2,TEMP,DTEMP,
     * FSPACE,TSPACE,FINCND,FSTOP,TSTOP,THCOEF
      COMMON /PUSHL/ SQRAT(20,20),SQRAZ(20,20),FMAS11(20,20),
     * FMAS22(20,20),FMAS23(20,20),FMAS33(20,20)
      LEVEL 2,SQRAT,SQRAZ,FMAS11,FMAS22,FMAS23,FMAS33
C
      COMMON /QLOGIC/ QAUX(20),QZETA,QSTRES,QPLAST,QSENS1,QEQUIL,
     * QDIAGN,QINGEO,QINVEL,QLOAD,QMATPR,QTHIKL,QTEMPE,QSPTEM,QAUXI1,
     * QAUXI2,QSPLOA,QIMPUL,QSHARP,QPESO,QIRCH,QSHEAR
C
      COMMON /TENCOM/ YY(3,2),YYY(3,2,2),A(3,3),B(3,3),AA(3,3),BB(3,3),
     * BM(3,3),DA(3,3),DB(3,3),G(3,3),GG(3,3),DN(3),DD(3,2),DDD(3,2,2),
     * DGAM(3,3),DGAMMX(3,3),HN(3,3),HQ(2),TAU(3,3),TAUSBL(3,3)
     * ,DD2(3,2),DDD2(3,2,2),YY2(3,2),YYY2(3,2,2),YYL(3,2),DBM(3,3)
C
      COMMON /THKNS/ Z,ZZ,ZCENTR,THICKN,ABSCIS(6,6),WEIGHT(6,6),
     * NGAUSS(4),IGAUSS,NLAYER,ILAYER,NSUBL(4),ISUBL
      COMMON /ONE/ DGAM33
      COMMON /GG/ SURFGG
C
      COMMON /DELC/ ICOUNT
C
      COMMON /STROUT/ ISB,L,LC,DISCR,TR(3,3),TN(3,3)
C
      COMMON /FRAC/ TAUF(3,3),TAUSPH,NUM,IPS1(10),IPS2(10)
C
      COMMON /STRPLO/ TAUP(60,3,3), TAUSPL(10,6)
      COMMON /MATRIX/ YLDFAC,ANUM
      COMMON /MATRIL/ LMAT(20,20,16),FMAT(6,20,20)
      LEVEL 2,LMAT,FMAT
C
C
      DIMENSION TAU11(JD1,JD2,JD3),TAU12(JD1,JD2,JD3),TAU13(JD1,JD2,JD3)
     * ,TAU21(JD1,JD2,JD3),TAU22(JD1,JD2,JD3),TAU23(JD1,JD2,JD3)
     * ,TAU31(JD1,JD2,JD3),TAU32(JD1,JD2,JD3),TAU33(JD1,JD2,JD3)
C
C
      IF (QSTRES) GO TO 20
      LEN=JD1*JD2*JD3
      CALL ERASE(TAU11,LEN)
      CALL ERASE(TAU12,LEN)
      CALL ERASE(TAU13,LEN)
      CALL ERASE(TAU21,LEN)
      CALL ERASE(TAU22,LEN)
      CALL ERASE(TAU23,LEN)
      CALL ERASE(TAU31,LEN)
```

```
      CALL ERASE(TAU32,LEN)
      CALL ERASE(TAU33,LEN)
      GSTRES=.TRUE.
C
   20 CONTINUE
      QQQ2=.FALSE.
      QQQ3=.FALSE.
      QQQ4=.FALSE.
      IF (ISTREZ.EQ.2) QQQ2=.TRUE.
      IF (ISTREZ.EQ.2.OR.ISTREZ.EQ.3) QQQ3=.TRUE.
      IF(ISTREZ .EQ. 4)QQQ4=.TRUE.
C
      DGTEMP=-(1.+HNU)*ALPHA*DTEMP/HNU
      DGAMMA=DGAMMX(1,1)+DGAMMX(2,2)+DGAMMX(3,3)+DGTEMP*3.
      IF(QQQ3 .OR. QQQ4) DGAMMA=DGAMMX(1,1)+DGAMMX(2,2)+DGTEMP*3.
      IF(HNU .EQ. 0.5) DGAMMA=3.*(DGTEMP+ALPHA*DTEMP)
      IF(HNU .EQ. 0.5) HNUP=HNU/3.
C
      ES=EE
      IF (BSTIV(ILAYER).NE.0.) CALL ESTIFF
      FACTOR=1.
      EEP=ES/(1.+HNU)
      HNUPP=1./3.
      IF(HNU .NE. 0.5) HNUP=HNU/(1.-2.*HNU)
      IF(QQQ3 .OR. QQQ4) HNUP=HNU/(1.-HNU)
      IF(QQQ3 .OR. QQQ4) HNUPP=(1.-2.*HNU)/(3.*(1.-HNU))
C
      IF(CONST.NE.0.) CALL SENSIT
C
      CALL ERASE(TAU,9)
      DGAM33=0.0
      NSBL=NSUBL(ILAYER)
C
C
      CALL ERASE(DTM,9)
      CALL ERASE (TAUF,9)
C       THE FOLLOWING T33,T33PL CALCULATION IS ONLY VALID FOR INITIALLY
C       CONSTANT THICKNESS PLATE
C
      IF(.NOT. QQQ4)GOTO 25
      T33=-P(I1,I2)*SURFGG*(1.+2.*Z/THICKN)/2.
      T33PL=-PPL(I1,I2)*SURFGG*(1.+2.*Z/THICKN)/2.
      DTAU33=G(3,3)*(T33PL-T33)
   25 CONTINUE
      IF(ICOUNT .EQ. 0)GOTO 1010
      IF(I1.EQ.IS1 .AND. I2.EQ.IS2)GOTO 1000
      GOTO 1010
 1000 III=ITIME/IOUT(12)*IOUT(12)-ITIME
      IF(III .NE. 0 .AND. .NOT.GPRINT(12))GOTO 1010
      WRITE(6,1007) P(I1,I2),PPL(I1,I2),SURFGG
 1007 FORMAT(' P(I1,I2)=',E22.15,' PPL(I1,I2)=',E22.15,
     *' SURFACE GG(3,3)=',E22.15)
 1010 CONTINUE
C       DETERMINE STRESSES FOR EACH SUBLAYER
      DO 250 ISB=1,NSBL
      QCZ=.FALSE.
      LI1I2=0
      L=1
      EL=1.
      SIGMSQ=SIGMA(ISB)*SIGMA(ISB)*FACTOR*FACTOR
      IZ=IZ+1
C
      CALL ERASE (TN,9)
      TN(1,1)=TAU11(I1,I2,IZ)
      TN(1,2)=TAU12(I1,I2,IZ)
      TN(2,1)=TAU21(I1,I2,IZ)
      TN(2,2)=TAU22(I1,I2,IZ)
      IF(QQQ2)GOTO 3050
      TN(1,3)=TAU13(I1,I2,IZ)
      TN(2,3)=TAU23(I1,I2,IZ)
      TN(3,1)=TAU31(I1,I2,IZ)
      TN(3,2)=TAU32(I1,I2,IZ)
      IF(QQQ3)GOTO 3050
      TN(3,3)=TAU33(I1,I2,IZ)
      IF(ITIME.GT.1 .AND. TN(3,3).EQ.0.0)T33=0.0
      IF(QQQ4)TN(3,3)=T33
 3050 CONTINUE
C
C       CALCULATE - TN(I,J) AT EACH SUBLAYER AND SUM
```

```
C
      DTM(1,1)=DTM(1,1)-TN(1,1)*CCEFF(ISB)
      DTM(1,2)=DTM(1,2)-TN(1,2)*CCEFF(ISB)
      DTM(2,1)=DTM(2,1)-TN(2,1)*CCEFF(ISB)
      DTM(2,2)=DTM(2,2)-TN(2,2)*CCEFF(ISB)
      IF (QQQ2) GO TO 5050
      DTM(1,3)=DTM(1,3)-TN(1,3)*CCEFF(ISB)
      DTM(2,3)=DTM(2,3)-TN(2,3)*CCEFF(ISB)
      DTM(3,1)=DTM(3,1)-TN(3,1)*CCEFF(ISB)
      DTM(3,2)=DTM(3,2)-TN(3,2)*CCEFF(ISB)
      DTM(3,3)=DTM(3,3)-TN(3,3)*CCEFF(ISB)
 5050 CONTINUE
C
C     TM(LA,LB) ARE THE MIXED INITIAL STRESSES FOR THIS PARTICULAR SUBLA
C     YER
C       SUBDIVIDED TIME STEP STARTS HERE, IF REQUIRED
   45 CONTINUE
      TM(1,1)=TN(1,1)
      TM(1,2)=TN(1,2)
      TM(1,3)=TN(1,3)
      TM(2,1)=TN(2,1)
      TM(2,2)=TN(2,2)
      TM(2,3)=TN(2,3)
      TM(3,1)=TN(3,1)
      TM(3,2)=TN(3,2)
      TM(3,3)=TN(3,3)
      TAUM=TM(1,1)+TM(2,2)+TM(3,3)
      LC=1
      IF(ICOUNT .EQ. 1) CALL AUXIL(4)
C
C      CALCULATE MIXED TRIAL ELASTIC STRESSES TR(LA,LB)
C
   50 CONTINUE
      HLAMDA=0.
      CALL ERASE(TR,9)
      ETERM1=EEP/EL
      ETERM2=HNUP*DGAMMA*ETERM1
      TR(1,1)=TM(1,1)+ETERM1*DGAMMX(1,1)+ETERM2
      TR(1,2)=TM(1,2)+ETERM1*DGAMMX(1,2)
      TR(2,1)=TM(2,1)+ETERM1*DGAMMX(2,1)
      TR(2,2)=TM(2,2)+ETERM1*DGAMMX(2,2)+ETERM2
      IF (QQQ2) GO TO 6050
      TR(1,3)=TM(1,3)+ETERM1*DGAMMX(1,3)
      TR(2,3)=TM(2,3)+ETERM1*DGAMMX(2,3)
      TR(3,1)=TM(3,1)+ETERM1*DGAMMX(3,1)
      TR(3,2)=TM(3,2)+ETERM1*DGAMMX(3,2)
      IF(QQQ3)GOTO 6050
      TR(3,3)=TM(3,3)+ETERM1*DGAMMX(3,3)+ETERM2
      IF(.NOT. QQQ4)GOTO 6050
      TR(1,1)=TR(1,1)+HNUP*DTAU33/EL
      TR(2,2)=TR(2,2)+HNUP*DTAU33/EL
      TR(3,3)=TM(3,3)+DTAU33/EL
 6050 CONTINUE
      TAUT=TR(1,1)+TR(2,2)+TR(3,3)
C
C     YIELD PHI
C
      CZ=TR(1,1)*TR(1,1)+2.*(TR(1,2)*TR(2,1)+TR(1,3)*TR(3,1)
     * +TR(2,3)*TR(3,2))+TR(2,2)*TR(2,2)+TR(3,3)*TR(3,3)
     * -(TAUT**2+2.*SIGMSQ)/3.
      IF(L .GT. LC)GOTO 55
      IF(ICOUNT .EQ. 1)CALL AUXIL(5)
C
C     TEST YIELD CONDITION
C
   55 IF(CZ.LT.0 .AND. L.EQ.LC)GOTO 220
      IF (CZ.LT.0.) GO TO 121
      IF(QQZ)GOTO 60
      L=INT(YLDFAC*(SQRT((1.5*CZ+SIGMSQ)/SIGMSQ)-1.0))+1
      EL=L
      QQZ=.TRUE.
      IF(L .GT. 1)GOTO 50
   60 CONTINUE
C
C     PLASTIC BEHAVIOR
C
C
C     MIXED CORRECTOR 3 DIMENSIONAL STRESSES TC(LA,LB) ARE INITIAL
C      DEVIATORIC STRESS (FOR THIS INCREMENT OR SUB-INCREMENT) BASED ON
C             TM(LA,LB)
C
```

```
      HTERM=HNUPP*TAUM
      CALL ERASE(TC,9)
      TC(1,1)=TM(1,1)-HTERM
      IF(QQQ4) TC(1,1)=TC(1,1)-HNUP*TM(3,3)
      TC(1,2)=TM(1,2)
      TC(2,1)=TM(2,1)
      TC(2,2)=TM(2,2)-HTERM
      IF(QQQ4) TC(2,2)=TC(2,2)-HNUP*TM(3,3)
      IF (QQQ2) GO TO 7050
      TC(1,3)=TM(1,3)
      TC(2,3)=TM(2,3)
      TC(3,1)=TM(3,1)
      TC(3,2)=TM(3,2)
      IF(QQQ3 .OR. QQQ4)GOTO 7050
      TC(3,3)=TM(3,3)-HTERM
 7050 CONTINUE
      TAUC=TC(1,1)+TC(2,2)+TC(3,3)
C
      IF (ISTRES .NE.1) GO TO 70
C   TANGENT APPROACH
      TAUM3=TAUM/3.
      HLAMDA=0.
      DGAMAT=DGAMMA/3.-ALPHA*DTEMP
      DO 65 LA=1,3
      DO 65 LB=1,3
   65 HLAMDA=HLAMDA+(TM(LA,LB)-TAUM3*DELTA(LA,LB))
     * *(DGAMMX(LB,LA)-DGAMAT*DELTA(LB,LA))
      HLAMDA=HLAMDA*EEP*1.5/(SIGMSQ*EL)
      IF (HLAMDA) 150,150,121
   70 CONTINUE
C      COMPUTE AND CHECK VALUES OF AZ,BZ AND DISCRIMINANT
C
      AZ=TC(1,1)*TC(1,1)+2.*(TC(1,2)*TC(2,1)+TC(1,3)*TC(3,1)
     * +TC(2,3)*TC(3,2))+TC(2,2)*TC(2,2)+TC(3,3)*TC(3,3)-(TAUC**2)/3.
      BZ=TC(1,1)*TR(1,1)+TC(1,2)*TR(2,1)+TC(1,3)*TR(3,1)
     * +TC(2,1)*TR(1,2)+TC(2,2)*TR(2,2)+TC(2,3)*TR(3,2)
     * +TC(3,1)*TR(1,3)+TC(3,2)*TR(2,3)+TC(3,3)*TR(3,3)-TAUC*TAUT/3.
      DISCR=BZ*BZ-AZ*CZ
      IF(L .GT. LI1I2) LI1I2=L
      IF(ICOUNT .EQ. 1)CALL AUXIL(6)
C
C     TEST AZ
C     IF AZ IS NEGATIVE - PRINT ERROR MESSAGE
C     IF AZ IS ZERO     - SUB-INCREMENT
C     IF AZ IS POSITIVE - CONTINUE
C
      IF(AZ) 80,150,100
   80 WRITE(NWRITE,90)
   90 FORMAT(1H ,4X,14HAA NEGATIVE AT )
      GO TO 180
C
C     TEST DISCRIMINANT
C
C     IF DISCR IS NEGATIVE - SUB-INCREMENT
C     OTHERWISE CONTINUE
  100 IF(DISCR.LT.0.) GO TO 150
C
C     TEST BZ
C
C     IF BZ IS NEGATIVE OR ZERO - SUB-INCREMENT
C     OTHERWISE CONTINUE
C
      IF(BZ.LE.0.)GOTO 150
C      COMPUTE HLAMDA AND ELASTOPLASTIC STRESSES
      HLAMD2= BZ+SQRT(DISCR)
      HLAMDA=CZ/HLAMD2
  121 CONTINUE
      TM(1,1)=TR(1,1)-HLAMDA*TC(1,1)
      TM(1,2)=TR(1,2)-HLAMDA*TC(1,2)
      TM(2,1)=TR(2,1)-HLAMDA*TC(2,1)
      TM(2,2)=TR(2,2)-HLAMDA*TC(2,2)
      IF (QQQ2) GO TO 8050
      TM(1,3)=TR(1,3)-HLAMDA*TC(1,3)
      TM(2,3)=TR(2,3)-HLAMDA*TC(2,3)
      TM(3,1)=TR(3,1)-HLAMDA*TC(3,1)
      TM(3,2)=TR(3,2)-HLAMDA*TC(3,2)
      IF(QQQ3)GOTO 8050
      TM(3,3)=TR(3,3)-HLAMDA*TC(3,3)
      IF(QQQ4) TM(3,3)=TR(3,3)
 8050 CONTINUE
```

```
      TAUM=TM(1,1)+TM(2,2)+TM(3,3)
      IF(ICOUNT .EQ. 1)CALL AUXIL(7)
C     CHECK THE SUB-INCREMENT NUMBER
C
      IF(LC.EQ.L)GOTO 210
      LC=LC+1
      GO TO 50
C
C     MAKE SUB-INCREMENTS SMALLER
C
  150 L=L+1
      EL=L
C
C     CHECK MAXIMUM NUMBER OF ALLOWABLE SUB-INCREMENTS
C
      IF(L-100) 45,45,160
  160 WRITE(NWRITE,170)
  170 FORMAT(1H ,4X,36HSTRESS CALCULATION UNSATISFACTORY AT )
  180 WRITE(NWRITE,190) ITIME,I1,I2,ILAYER,ISB,L,LC
  190 FORMAT(1H ,9X,9HTIME STEP,I6,5X,3HI1=,I2,5X,3HI2=,I2,5X,6HLAYER=,I
     *2,9HSUBLAYER=,I2,5X,2HL=,I3,5X,3HLC=,I3)
      WRITE(NWRITE,200) AZ,BZ,CZ,DISCR,
     * ((TM(IV,JV),JV=1,3),IV=1,3),((TR(IV,JV),
     * JV=1,3),IV=1,3),((TC(IV,JV),JV=1,3),IV=1,3)
  200 FORMAT(10X,"AZ     =",E15.6,10X,"BZ     =",E15.6,10X,"CZ     =",E15.6
     *,10X,"DISCR =",E15.6
     */10X,"MIXED INITIAL STRESSES",16X,"=  TM  =",3E15.6/56X,3E15.6/
     *56X,3E15.6/10X,"MIXED TRIAL ELASTIC STRESSES",10X,"=  TR  =",
     *3E15.6/56X,3E15.6/56X,3E15.6/10X,"MIXED CORRECTOR 3-DIMENSIONAL ST
     *RESSES=  TC  =",3E15.6/56X,3E15.6/56X,3E15.6)
      CALL DIAGNO(5)
C
C      HAVE REACHED PLASTIC SOLUTION
C
  210 CONTINUE
      IPLAST(I1,I2)=IPLAST(I1,I2)+1
      GO TO 222
  220 CONTINUE
C
C      ELASTIC TM EQUALS TRIAL TR PER SUBLAYER
C
      TM(1,1)=TR(1,1)
      TM(1,2)=TR(1,2)
      TM(2,1)=TR(2,1)
      TM(2,2)=TR(2,2)
      IF (QQQ2) GO TO 9050
      TM(1,3)=TR(1,3)
      TM(2,3)=TR(2,3)
      TM(3,1)=TR(3,1)
      TM(3,2)=TR(3,2)
      TM(3,3)=TR(3,3)
 9050 CONTINUE
C
  222 CONTINUE
      DGAMMX(3,3)=((TM(3,3)-TN(3,3))-HNU*((TM(1,1)-TN(1,1))+(TM(2,2)-TN(
     * 2,2))))/ES+(1.+HNU)*(2.*TN(3,3)-TN(1,1)-TN(2,2))*HLAMDA/(3.*ES)
      IF(ICOUNT .EQ. 1)CALL AUXIL (9)
C
C       STORE MIXED TENSOR SUBLAYER STRESSES
      IF(ICOUNT .EQ. 0)GOTO 2500
C
      TAU11(I1,I2,IZ)=TM(1,1)
      TAU12(I1,I2,IZ)=TM(1,2)
      TAU21(I1,I2,IZ)=TM(2,1)
      TAU22(I1,I2,IZ)=TM(2,2)
      IF(QQQ2)GOTO 2500
      TAU13(I1,I2,IZ)=TM(1,3)
      TAU23(I1,I2,IZ)=TM(2,3)
      TAU31(I1,I2,IZ)=TM(3,1)
      TAU32(I1,I2,IZ)=TM(3,2)
      IF(QQQ3)GOTO 2500
      TAU33(I1,I2,IZ)=TM(3,3)
 2500 CONTINUE
C
C
C      CALLCULATE TOTAL MIXED TENSOR STRESSES FROM SUBLAYER STRESSES
C
      LMAT(I1,I2,IZ)=LI1I2
      TAUF(1,1)=TAUF(1,1)+COEFF(ISB)*TM(1,1)
      TAUF(1,2)=TAUF(1,2)+COEFF(ISB)*TM(1,2)
```

```
      TAUF(2,1)=TAUF(2,1)+COEFF(ISB)*TM(2,1)
      TAUF(2,2)=TAUF(2,2)+COEFF(ISB)*TM(2,2)
      IF(QQQ2)GOTO 2501
      TAUF(1,3)=TAUF(1,3)+COEFF(ISB)*TM(1,3)
      TAUF(3,1)=TAUF(3,1)+COEFF(ISB)*TM(3,1)
      TAUF(2,3)=TAUF(2,3)+COEFF(ISB)*TM(2,3)
      TAUF(3,2)=TAUF(3,2)+COEFF(ISB)*TM(3,2)
      IF(QQQ3)GOTO 2501
      TAUF(3,3)=TAUF(3,3)+COEFF(ISB)*TM(3,3)
 2501 CONTINUE
      DGAM33=DGAM33+COEFF(ISB)*DGAMMX(3,3)
      ISUBL=ISB
      IF(ICOUNT .EQ. 1)CALL AUXIL(3)
C
  250 CONTINUE
C                CONTRAVARIANT TOTAL STRESSES
      TAU(1,1)=TAUF(1,1)*GG(1,1)+TAUF(1,2)*GG(2,1)+TAUF(1,3)*GG(3,1)
      TAU(1,2)=TAUF(1,1)*GG(1,2)+TAUF(1,2)*GG(2,2)+TAUF(1,3)*GG(3,2)
      TAU(2,1)=TAUF(2,1)*GG(1,1)+TAUF(2,2)*GG(2,1)+TAUF(2,3)*GG(3,1)
      TAU(2,2)=TAUF(2,1)*GG(1,2)+TAUF(2,2)*GG(2,2)+TAUF(2,3)*GG(3,2)
      TAU(1,2)=.5*(TAU(1,2)+TAU(2,1))
      TAU(2,1)=TAU(1,2)
      IF(QQQ2)GOTO 2600
      TAU(1,3)=TAUF(1,1)*GG(1,3)+TAUF(1,2)*GG(2,3)+TAUF(1,3)*GG(3,3)
      TAU(3,1)=TAUF(3,1)*GG(1,1)+TAUF(3,2)*GG(2,1)+TAUF(3,3)*GG(3,1)
      TAU(2,3)=TAUF(2,1)*GG(1,3)+TAUF(2,2)*GG(2,3)+TAUF(2,3)*GG(3,3)
      TAU(3,2)=TAUF(3,1)*GG(1,2)+TAUF(3,2)*GG(2,2)+TAUF(3,3)*GG(3,2)
      TAU(3,3)=TAUF(3,1)*GG(1,3)+TAUF(3,2)*GG(2,3)+TAUF(3,3)*GG(3,3)
      TAU(1,3)=.5*(TAU(1,3)+TAU(3,1))
      TAU(3,1)=TAU(1,3)
      TAU(2,3)=.5*(TAU(2,3)+TAU(3,2))
      TAU(3,2)=TAU(2,3)
 2600 CONTINUE
      IF(ICOUNT .EQ. 0)GOTO 270
      ANUM=1HE
      IF(IPLAST(I1,I2) .GT. 0) ANUM=1HP
      TAUSPH=(TAUF(1,1)+TAUF(2,2)+TAUF(3,3))/3.0
      IF(MAUXIL .EQ. 1) CALL MAXMIN (TAUF,1)
      CALL AUXIL (11)
      DO 260 LL=1,NUM
      IF(I1 .NE. IPS1(LL))GOTO 260
      IF(I2 .NE. IPS2(LL))GOTO 260
      TAUSPL(LL,IGAUSS)=TAUSPH
      LZ=IGAUSS+NGAUSS(ILAYER)*(LL-1)*ILAYER
      DO 255 I=1,3
      DO 255 J=1,3
      TAUP(LZ,I,J)=TAUF(I,J)
  255 CONTINUE
  260 CONTINUE
C
      FMAT(IGAUSS,I1,I2)=ANUM
  270 CONTINUE
C       CALCULATE CHANGE IN TM PER GAUSS STATION AS TM - TN
C
      DTM(1,1)=DTM(1,1)+TAU(1,1)*G(1,1)+TAU(1,2)*G(2,1)+TAU(1,3)*G(3,1)
      DTM(1,2)=DTM(1,2)+TAU(1,1)*G(1,2)+TAU(1,2)*G(2,2)+TAU(1,3)*G(3,2)
      DTM(2,1)=DTM(2,1)+TAU(2,1)*G(1,1)+TAU(2,2)*G(2,1)+TAU(2,3)*G(3,1)
      DTM(2,2)=DTM(2,2)+TAU(2,1)*G(1,2)+TAU(2,2)*G(2,2)+TAU(2,3)*G(3,2)
      IF (QQQ2) GO TO 9070
      DTM(1,3)=DTM(1,3)+TAU(1,1)*G(1,3)+TAU(1,2)*G(2,3)+TAU(1,3)*G(3,3)
      DTM(2,3)=DTM(2,3)+TAU(2,1)*G(1,3)+TAU(2,2)*G(2,3)+TAU(2,3)*G(3,3)
      DTM(3,1)=DTM(3,1)+TAU(3,1)*G(1,1)+TAU(3,2)*G(2,1)+TAU(3,3)*G(3,1)
      DTM(3,2)=DTM(3,2)+TAU(3,1)*G(1,2)+TAU(3,2)*G(2,2)+TAU(3,3)*G(3,2)
      DTM(3,3)=DTM(3,3)+TAU(3,1)*G(1,3)+TAU(3,2)*G(2,3)+TAU(3,3)*G(3,3)
 9070 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ZETA
C
C     EVALUATE STRESSES THROUGH THE THICKNESS AND EVALUATE
C     CONTRAVARIANT COMPONENTS OF THE RELATIVE MOMENT RESULTANT TENSOR
C     AND OF THE RELATIVE STRESS RESULTANT TENSOR
C
      IMPLICIT LOGICAL(Q)
C
      COMMON/ ALLENE /TOTAL,TOTKIN,TOTELA,TOTPLA,TOTWEX,TOTTEM,INERGY
     * , TOTVIS,TOTE1,TOTE2 ,ECHECK
     * ,DTM(3,3),SQRG,SQRA,SIGMSQ,AZ,BZ,CZ
C
      COMMON /APTRAC/ KSPOTS,KT(4),KDIR(4,3),MMTYPE(4),MTTYPE(4),KSPOT,
     * KSIDE,STIFN(4,3),TORQK(4)
      COMMON /APTRAL/ TA1(3,80),TA2(3,80),HM11(80),HM22(80)
      LEVEL 2,TA1,TA2,HM11,HM22
C
      COMMON /CARTE/ YTEST,YNEW,YSAVE
      COMMON /CARTEL/ Y(3,20,20),D(3,20,20),Y2(3,20,20),D2(3,20,20)
      LEVEL 2,Y,D,Y2,D2
C
      COMMON /CTIME/ AUX(20),TIME,DELTAT,TIMEF,ITIME,ITIMEF,IAUX(20),
     * IQUT(20),QPRINT(20)
      COMMON /CTIMEL/ IPLAST(20,20),P(20,20),PPL(20,20)
      LEVEL 2,IPLAST,P,PPL
C
      COMMON/CTHKN/ ZA(20,20),ZB(20,20),DZA1(20,20),DZA2(20,20),
     * DZB1(20,20),DZB2(20,20),BBAR(80)
      LEVEL 2,ZA,ZB,DZA1,DZA2,DZB1,DZB2,BBAR
C
      COMMON /INDEX/ NREAD,NWRITE,NPUNCH,NMESH1,NMESH2,N1,N2,NZ,N1M,N2M,
     * N1MM,N2MM,I1,I2,IZ,I1ZERO,I2ZERO,IRY1,IRY2,IRY3,IRY4,ISTR1,ISTR2,
     * ISTR3,ISTR4,IC1,IC2,ID1,ID2,IP1,IP2,IS1,IS2,K1,K2,K3,K4,KRUN,
     * KZSTOP,KYTEST,IDIR,I1TEST,I2TEST,KINITL
C
      COMMON /NEW1/ GDISP,MDISP,IST1,IST2,IST3,IST4,QTRAC,MTRAC,
     * TAUMAX,TAUMIN,GAMMAX,GAMMIN,ELMAX,ELMIN,I1MAXS,I2MAXS,IGAMAS,
     * ILAMAS,I1MINS,I2MINS,IGAMIS,ILAMIS,I1MAXX,I2MAXX,IZMAXX,I1MINX,
     * I2MINX,IZMINX,I1PMAX,I2PMAX,IZPMAX,I1PMIN,I2PMIN,IZPMIN,III(20),
     * QTRACT,QTRACM,IPHY(20),MPHYS
C
      COMMON /OPTION/ MAUXIL,MINGEO,MINVEL,MLOAD,MMATPR,MSPLCA,
     * MSPTEM,MTEMPE,MTHIKL,MIMPUL,ISTRES,INCRML,ISTREZ
C
      COMMON /PHYSCN/ EE,HNU,ALPHA,CONST,EXPON,FACTOR,RATE,RHO,
     * HLAMDA,COEFF(5),SIGMA(5),TM(3,3),TC(3,3),DELTA(3,3)
C
      COMMON /PUSH/ FORCES(3),VELOC(3),RATIO,RATIOM,DX1,DX2,TEMP,DTEMP,
     * FSPACE,TSPACE,FINCNO,FSTOP,TSTOP,THCOEF
      COMMON /PUSHL/ SQRAT(20,20),SQRAZ(20,20),FMAS11(20,20),
     * FMAS22(20,20),FMAS23(20,20),FMAS33(20,20)
      LEVEL 2,SQRAT,SQRAZ,FMAS11,FMAS22,FMAS23,FMAS33
C
      COMMON /QLOGIC/ QAUX(20),QZETA,QSTRES,QPLAST,QSENS1,QEQUIL,
     * QDIAGN,QINGEO,QINVEL,QLOAD,QMATPR,QTHIKL,QTEMPE,QSPTEM,QAUXI1,
     * QAUXI2,QSPLOA,QIMPUL,QSHARP,QPESD,QIRCH,QSHEAR
C
      COMMON /SURNOM/ SNPR(3)
      COMMON /SURNOL/ SN(3,20,20)
      LEVEL 2,SN
C
      COMMON /S2/ STRESE(3)
      COMMON /S2L/ STRESL(3,20,20),STRESQ(2,20,20),STRESP(2,20,20)
      LEVEL 2,STRESL,STRESQ,STRESP
C
      COMMON /TENCOM/ YY(3,2),YYY(3,2,2),A(3,3),B(3,3),AA(3,3),BB(3,3),
     * BM(3,3),DA(3,3),DB(3,3),G(3,3),GG(3,3),DN(3),CD(3,2),CCC(3,2,2),
     * DGAM(3,3),DGAMMX(3,3),HN(3,3),HC(2),TAU(3,3),TAUSBL(3,3)
     * ,DD2(3,2),DDD2(3,2,2),YY2(3,2),YYY2(3,2,2),YYU(3,2),DBM(3,3)
C
      COMMON /THKNS/ Z,ZZ,ZCENTR,THICKN,ABSCIS(6,6),WEIGHT(6,6),
     * NGAUSS(4),IGAUSS,NLAYER,ILAYER,NSUBL(4),ISUBL
C
      COMMON /TNCOMP/ HM1(2,20,20),HM2(2,20,20),CAPQ1(2,20,20),
     * CAPQ2(2,20,20),CAPQ3(2,20,20),CAP2Q1(2,20,20),
     * CAP2Q2(2,20,20),CAP2Q3(2,20,20)
      LEVEL 2,HM1,HM2,CAPQ1,CAPQ2,CAPQ3,CAP2Q1,CAP2Q2,CAP2Q3
C
      COMMON /VARTHK/ VTHIK(4,20,20),CENTRV(4,20,20)
      LEVEL 2,VTHIK,CENTRV
```

```
C
      COMMON /VIS1/ IVIS,AVIS(4),GVIS(4),DSTR11,DSTR12,DSTR13,DSTR21,
     * DSTR22,DSTR23,DSTR31,DSTR32,DSTR33
      COMMON /VIS1L/ VSTR11(4,20,20),VSTR12(4,20,20),VSTR13(4,20,20),
     * VSTR22(4,20,20),VSTR23(4,20,20)
      LEVEL 2,VSTR11,VSTR12,VSTR13,VSTR22,VSTR23
C
      COMMON /STRINT/ EPSL1(20,20),EPSL2(20,20),GAMMAL(20,20),
     * EPSU1(20,20),EPSU2(20,20),GAMMAU(20,20)
      LEVEL 2,EPSL1,EPSL2,GAMMAL,EPSU1,EPSU2,GAMMAU
      COMMON /MAI1/ NCONT,NRITE,NTRAIN,MRITE,NCELP,ETAD1,ETAD2,NSTRN
     * ,FACTOM,MQ,FACTON,NQ,FKIN
      COMMON /ONE/ OGAM33
      COMMON /THREE/ AGAM33
      COMMON /TEMP/ PGAM33
C
      COMMON /DELC/ ICOUNT
      COMMON /DEL/ DELBAR(20,20)
      LEVEL 2,DELBAR
      COMMON /GG/SURFGG
      COMMON /GPRINT/ GTYPE,GBASE(3,3),DGBASE(3,3)
C
      COMMON /FRAC/ TAUF(3,3),TAUSPH,NUM,IPS1(10),IPS2(10)
C
      DIMENSION DZ(3),CS(3,2),CX(3,3),HM(2,2)
C
      PAR=0.
C     FOR ORTHOGONAL COORDINATES, TO REDUCE ERROR
C
      IF(K1.NE.7.AND.K2.NE.7) GO TO 10
      DA(1,2)=0.
      DA(2,1)=0.
      DB(1,2)=0.
      DB(2,1)=0.
   10 CONTINUE
      CALL ERASE(HM,4)
      STRESL(1,I1,I2)=0.
      STRESL(2,I1,I2)=0.
      STRESL(3,I1,I2)=0.
      CAPQ1(1,I1,I2)=0.
      CAPQ1(2,I1,I2)=0.
      CAPQ2(1,I1,I2)=0.
      CAPQ2(2,I1,I2)=0.
      CAPQ3(1,I1,I2)=0.
      CAPQ3(2,I1,I2)=0.
      CAP2Q1(1,I1,I2)=0.
      CAP2Q1(2,I1,I2)=0.
      CAP2Q2(1,I1,I2)=0.
      CAP2Q2(2,I1,I2)=0.
      CAP2Q3(1,I1,I2)=0.
      CAP2Q3(2,I1,I2)=0.
      STRESP(1,I1,I2)=0.
      STRESP(2,I1,I2)=0.
      STRESQ(1,I1,I2)=0.
      STRESQ(2,I1,I2)=0.
      DZ(1)=0.
      DZ(2)=0.
      DZ(3)=0.
      ZZ=0.
      DB(3,3)=0.
      IPLAST(I1,I2)=0
C     MID-SURFACE OFFSET
C     FOR KIRCHHOFF OR SHEAR SHELL CONSIDER DEL=0 SINCE, FROM THIKLA,
C     THE POINT Z=0 IS THE HALF THICKNESS OF THE SHELL AND IS NOT WEIGHE
C     VARIATION IN THE DENSITY
      DEL=0.0
      IF (QIRCH .OR.QSHEAR) GO TO 2205
      THIKZ=0.
      DO 2204 ILAYER =1,3
      CALL THIKLA(1)
 2204 THIKZ=THIKZ+THICKN
      ZCEN=THIKZ*.5
      ILAYER=1
      CALL THIKLA(1)
      ZZCEN=THICKN-ZB(I1,I2)
      DEL=ZCEN-ZZCEN
 2205 CONTINUE
      CS(1,1)=BM(1,1)*YY(1,1)+BM(2,1)*YY(1,2)
      CS(1,2)=BM(1,2)*YY(1,1)+BM(2,2)*YY(1,2)
      CS(2,1)=BM(1,1)*YY(2,1)+BM(2,1)*YY(2,2)
      CS(2,2)=BM(1,2)*YY(2,1)+BM(2,2)*YY(2,2)
      CS(3,1)=BM(1,1)*YY(3,1)+BM(2,1)*YY(3,2)
```

```
      CS(3,2)=BM(1,2)*YY(3,1)+BM(2,2)*YY(3,2)
      CALL ERASE (HN,9)
      HM1(1,I1,I2)=0.0
      HM1(2,I1,I2)=0.0
      HM2(1,I1,I2)=0.0
      HM2(2,I1,I2)=0.0
      IZ=0
      IF(ICOUNT .EQ. 1) CALL ENERGY(2,1.0)
      IF(INORML.EQ.2) DGAMA3=0.
      IF(INORML.EQ.2) THICKZ=0.
      DGM33=0.0
      SUMG=0.0
      THIC=0.0
      SGAM33=0.0
C      CALCULATIONS FOR ALL LAYERS
      DO 169 ILAYER=1,NLAYER
      ISTREZ=ISTRES
      IF (QIRCH) GO TO 63
      IF (NLAYER.EQ.1) GO TO 61
      IF(ILAYER-2) 60,61,62
   60 ZZ=ZB(I1,I2)
      DZ(1)=DZB1(I1,I2)
      DZ(2)=DZB2(I1,I2)
      DZ(3)=0.
      GO TO 63
   61 CONTINUE
      IF(.NOT.QIRCH.AND..NOT.QSHEAR.AND.ISTRES.EQ.2) ISTREZ=3
      DZ(1)=0.
      DZ(2)=0.
      DZ(3)=1.
      GO TO 63
   62 ZZ=ZA(I1,I2)
      DZ(1)=DZA1(I1,I2)
      DZ(2)=DZA2(I1,I2)
      DZ(3)=0.
   63 CONTINUE
      CALL THIKLA(1)
      IF(INORML.EQ.2) THICKZ=THICKZ+THICKN
      THIC=THIC+THICKN
      NGAUSL=NGAUSS(ILAYER)
C           SURFACE TRAXON
      CALL SFORCE (CS,THICKN,YY,YY2,I1,I2,SURFGG)
C      CALCULATIONS FOR EACH GAUSS POINT
      DO 169 IGAUSS=1,NGAUSL
C     Z AS MEASURED FROM MID-SURFACE
      Z=.5*THICKN*ABSCIS(IGAUSS,NGAUSL)+ZCENTR
C     Z AS MEASURED FROM REFERENCE SURFACE
      Z=Z+DEL
      IF (ILAYER.EQ.2.OR.(.NOT.QIRCH.AND.NLAYER.EQ.1)) ZZ=Z
      IF (QSHEAR) ZZ=Z
      IF (MTEMPE .NE. 0) CALL TEMPER
      IF (MTEMPE .NE. 0 .OR. IGAUSS .EQ. 1) CALL MATPRO
      IF(MTEMPE.NE.0 .OR. IGAUSS.EQ.1) CALL ENERGY(6,1.0)
      GBTN=0.0
C      CALCULATIONS FOR ALL COMPONENTS
      DO 9 J=1,3
      GBASE(3,J)=SN(J,I1,I2)
      IF(.NOT.QSHEAR.AND.INORML.EQ.2) GBASE(3,J)=GBASE(3,J)+Y2(J,I1,I2)
      IF (QIRCH) GO TO 9891
      IF(.NOT.QSHEAR.AND.INORML.EQ.2) GO TO 9891
      GBASE(3,J)=GBASE(3,J) +DZ(3)*Y2(J,I1,I2)
 9891 CONTINUE
      GBTN=GBTN+GBASE(3,J)*SN(J,I1,I2)
      GBASE(1,J)=YY(J,1)-Z*CS(J,1)
      GBASE(2,J)=YY(J,2)-Z*CS(J,2)
      IF(QIRCH) GO TO 9
      GBASE(1,J)=GBASE(1,J)+DZ(1)*Y2(J,I1,I2)+ZZ*YY2(J,1)
      GBASE(2,J)=GBASE(2,J)+DZ(2)*Y2(J,I1,I2)+ZZ*YY2(J,2)
    9 CONTINUE
      CALL ERASE(DGBASE,9)
      DGBASE(3,1)=DN(1)
      DGBASE(3,2)=DN(2)
      DGBASE(3,3)=DN(3)
      IF(QIRCH) GO TO 886
      DGBASE(3,1)=DGBASE(3,1)+DZ(3)*D2(1,I1,I2)
      DGBASE(3,2)=DGBASE(3,2)+DZ(3)*D2(2,I1,I2)
      DGBASE(3,3)=DGBASE(3,3)+DZ(3)*D2(3,I1,I2)
  886 CONTINUE
```

```
      DO 888 LA=1,2
      DO 888 K=1,3
      DO 887 LS=1,2
  887 DGBASE(LA,K)=DGBASE(LA,K)+(DELTA(LS,LA)-Z*BM(LS,LA))*DD(K,LS)-
     * Z*DBM(LS,LA)*(YY(K,LS)-DD(K,LS))
      IF (QIRCH) GO TO 888
      DGBASE(LA,K)=DGBASE(LA,K)+ZZ*DD2(K,LA)+DZ(LA)*D2(K,I1,I2)
  888 CONTINUE
      G(1,1)=GBASE(1,1)*GBASE(1,1)+GBASE(1,2)*GBASE(1,2)
     * +GBASE(1,3)*GBASE(1,3)
      G(1,2)=GBASE(1,1)*GBASE(2,1)+GBASE(1,2)*GBASE(2,2)
     * +GBASE(1,3)*GBASE(2,3)
      G(1,3)=GBASE(1,1)*GBASE(3,1)+GBASE(1,2)*GBASE(3,2)
     * +GBASE(1,3)*GBASE(3,3)
      G(2,1)=G(1,2)
      G(2,2)=GBASE(2,1)*GBASE(2,1)+GBASE(2,2)*GBASE(2,2)
     * +GBASE(2,3)*GBASE(2,3)
      G(2,3)=GBASE(2,1)*GBASE(3,1)+GBASE(2,2)*GBASE(3,2)
     * +GBASE(2,3)*GBASE(3,3)
      G(3,1)=G(1,3)
      G(3,2)=G(2,3)
      G(3,3)=GBASE(3,1)*GBASE(3,1)+GBASE(3,2)*GBASE(3,2)
     * +GBASE(3,3)*GBASE(3,3)
      CALL ERASE(DGAM,9)
      IJ=3
      IF(QIRCH) IJ=2
      DO 102 I=1,IJ
      DO 102 J=1,IJ
      DO 102 K=1,3
  102 DGAM(I,J)=DGAM(I,J)+.5*(GBASE(I,K)*DGBASE(J,K)+GBASE(J,K)*DGBASE(
     * I,K)-DGBASE(I,K)*DGBASE(J,K))
C     G=GTYPE
      GTYPE=G(1,1)*(G(2,2)*G(3,3)-G(2,3)**2)+G(1,2)*(G(1,3)*G(2,3)-
     * G(1,2)*G(3,3))+G(1,3)*(G(1,2)*G(2,3)-G(1,3)*G(2,2))
      SQRG=SQRT(GTYPE)
      GTYPE=1./GTYPE
      GG(1,1)=(G(2,2)*G(3,3)-G(2,3)**2)*GTYPE
      GG(1,2)=(G(1,3)*G(2,3)-G(1,2)*G(3,3))*GTYPE
      GG(2,1)=GG(1,2)
      GG(1,3)=(G(1,2)*G(2,3)-G(1,3)*G(2,2))*GTYPE
      GG(3,1)=GG(1,3)
      GG(2,2)=(G(1,1)*G(3,3)-G(1,3)**2)*GTYPE
      GG(2,3)=(G(1,2)*G(1,3)-G(1,1)*G(3,2))*GTYPE
      GG(3,2)=GG(2,3)
      GG(3,3)=(G(1,1)*G(2,2)-G(1,2)**2)*GTYPE
C
C
      DGAMMX(1,1)=DGAM(1,1)*GG(1,1)+DGAM(1,2)*GG(2,1)+DGAM(1,3)*GG(3,1)
      DGAMMX(1,2)=DGAM(2,1)*GG(1,1)+DGAM(2,2)*GG(2,1)+DGAM(2,3)*GG(3,1)
      DGAMMX(1,3)=DGAM(3,1)*GG(1,1)+DGAM(3,2)*GG(2,1)+DGAM(3,3)*GG(3,1)
      DGAMMX(2,1)=DGAM(1,1)*GG(1,2)+DGAM(1,2)*GG(2,2)+DGAM(1,3)*GG(3,2)
      DGAMMX(2,2)=DGAM(2,1)*GG(1,2)+DGAM(2,2)*GG(2,2)+DGAM(2,3)*GG(3,2)
      DGAMMX(2,3)=DGAM(3,1)*GG(1,2)+DGAM(3,2)*GG(2,2)+DGAM(3,3)*GG(3,2)
      DGAMMX(3,1)=DGAM(1,1)*GG(1,3)+DGAM(1,2)*GG(2,3)+DGAM(1,3)*GG(3,3)
      DGAMMX(3,2)=DGAM(2,1)*GG(1,3)+DGAM(2,2)*GG(2,3)+DGAM(2,3)*GG(3,3)
      DGAMMX(3,3)=DGAM(3,1)*GG(1,3)+DGAM(3,2)*GG(2,3)+DGAM(3,3)*GG(3,3)
      D33S=DGAMMX(3,3)
      IF(ICOUNT .EQ. 1)CALL AUXIL(8)
C     THIS IS A CALL TO SUBROUTINE VISCUS
      IF(AVIS(ILAYER).NE. 0.) CALL DIMTAU(3,0)
C     THIS IS A CALL TO SUBROUTINE STRESS
      CALL DIMTAU(2,0)
      IF(MPHYS.NE.0 .AND. ICOUNT.EQ.1)CALL PHYSIC(2)
      PAR=THICKN*.5*WEIGHT(IGAUSS,NGAUSL)
      PARSQ=PAR*SQRG
      HN(1,1)=HN(1,1)+PARSQ*(TAU(1,1)*GBASE(1,1)+TAU(1,2)*GBASE(2,1)
     * +TAU(1,3)*GBASE(3,1))
      HN(1,2)=HN(1,2)+PARSQ*(TAU(1,1)*GBASE(1,2)+TAU(1,2)*GBASE(2,2)
     * +TAU(1,3)*GBASE(3,2))
      HN(1,3)=HN(1,3)+PARSQ*(TAU(1,1)*GBASE(1,3)+TAU(1,2)*GBASE(2,3)
     * +TAU(1,3)*GBASE(3,3))
      HN(2,1)=HN(2,1)+PARSQ*(TAU(2,1)*GBASE(1,1)+TAU(2,2)*GBASE(2,1)
     * +TAU(2,3)*GBASE(3,1))
      HN(2,2)=HN(2,2)+PARSQ*(TAU(2,1)*GBASE(1,2)+TAU(2,2)*GBASE(2,2)
     * +TAU(2,3)*GBASE(3,2))
      HN(2,3)=HN(2,3)+PARSQ*(TAU(2,1)*GBASE(1,3)+TAU(2,2)*GBASE(2,3)
     * +TAU(2,3)*GBASE(3,3))
      IF (QIRCH) GO TO 2165
      PRSQD1=PARSQ*DZ(1)
      PRSQD2=PARSQ*DZ(2)
```

```
      PRSQD3=PARSQ*DZ(3)
      STRESL(1,I1,I2)=STRESL(1,I1,I2)+
     * PRSQD1*
     * (TAU(1,1)*GBASE(1,1)+TAU(1,2)*GBASE(2,1)+TAU(1,3)*GBASE(3,1))+
     * PRSQD2*
     * (TAU(2,1)*GBASE(1,1)+TAU(2,2)*GBASE(2,1)+TAU(2,3)*GBASE(3,1))+
     * PRSQD3*
     * (TAU(3,1)*GBASE(1,1)+TAU(3,2)*GBASE(2,1)+TAU(3,3)*GBASE(3,1))
      STRESL(2,I1,I2)=STRESL(2,I1,I2)+
     * PRSQD1*
     * (TAU(1,1)*GBASE(1,2)+TAU(1,2)*GBASE(2,2)+TAU(1,3)*GBASE(3,2))+
     * PRSQD2*
     * (TAU(2,1)*GBASE(1,2)+TAU(2,2)*GBASE(2,2)+TAU(2,3)*GBASE(3,2))+
     * PRSQD3*
     * (TAU(3,1)*GBASE(1,2)+TAU(3,2)*GBASE(2,2)+TAU(3,3)*GBASE(3,2))
      STRESL(3,I1,I2)=STRESL(3,I1,I2)+
     * PRSQD1*
     * (TAU(1,1)*GBASE(1,3)+TAU(1,2)*GBASE(2,3)+TAU(1,3)*GBASE(3,3))+
     * PRSQD2*
     * (TAU(2,1)*GBASE(1,3)+TAU(2,2)*GBASE(2,3)+TAU(2,3)*GBASE(3,3))+
     * PRSQD3*
     * (TAU(3,1)*GBASE(1,3)+TAU(3,2)*GBASE(2,3)+TAU(3,3)*GBASE(3,3))
      PRSQZZ=PARSQ*ZZ
      CAP2Q1(1,I1,I2)=CAP2Q1(1,I1,I2)+PRSQZZ*(TAU(1,1)*GBASE(1,1)
     * +TAU(1,2)*GBASE(2,1)+TAU(1,3)*GBASE(3,1))
      CAP2Q1(2,I1,I2)=CAP2Q1(2,I1,I2)+PRSQZZ*(TAU(2,1)*GBASE(1,1)
     * +TAU(2,2)*GBASE(2,1)+TAU(2,3)*GBASE(3,1))
      CAP2Q2(1,I1,I2)=CAP2Q2(1,I1,I2)+PRSQZZ*(TAU(1,1)*GBASE(1,2)
     * +TAU(1,2)*GBASE(2,2)+TAU(1,3)*GBASE(3,2))
      CAP2Q2(2,I1,I2)=CAP2Q2(2,I1,I2)+PRSQZZ*(TAU(2,1)*GBASE(1,2)
     * +TAU(2,2)*GBASE(2,2)+TAU(2,3)*GBASE(3,2))
      CAP2Q3(1,I1,I2)=CAP2Q3(1,I1,I2)+PRSQZZ*(TAU(1,1)*GBASE(1,3)
     * +TAU(1,2)*GBASE(2,3)+TAU(1,3)*GBASE(3,3))
      CAP2Q3(2,I1,I2)=CAP2Q3(2,I1,I2)+PRSQZZ*(TAU(2,1)*GBASE(1,3)
     * +TAU(2,2)*GBASE(2,3)+TAU(2,3)*GBASE(3,3))
 2165 CONTINUE
      IF (QSHEAR) GO TO 2059
      CX(1,1)=GBASE(1,1)*YYU(1,1)+GBASE(1,2)*YYU(2,1)
     * +GBASE(1,3)*YYU(3,1)
      CX(1,2)=GBASE(2,1)*YYU(1,1)+GBASE(2,2)*YYU(2,1)
     * +GBASE(2,3)*YYU(3,1)
      CX(1,3)=GBASE(3,1)*YYU(1,1)+GBASE(3,2)*YYU(2,1)
     * +GBASE(3,3)*YYU(3,1)
      CX(2,1)=GBASE(1,1)*YYU(1,2)+GBASE(1,2)*YYU(2,2)
     * +GBASE(1,3)*YYU(3,2)
      CX(2,2)=GBASE(2,1)*YYU(1,2)+GBASE(2,2)*YYU(2,2)
     * +GBASE(2,3)*YYU(3,2)
      CX(2,3)=GBASE(3,1)*YYU(1,2)+GBASE(3,2)*YYU(2,2)
     * +GBASE(3,3)*YYU(3,2)
      CX(3,1)=GBASE(1,1)*SN(1,I1,I2)+GBASE(1,2)*SN(2,I1,I2)
     * +GBASE(1,3)*SN(3,I1,I2)
      CX(3,2)=GBASE(2,1)*SN(1,I1,I2)+GBASE(2,2)*SN(2,I1,I2)
     * +GBASE(2,3)*SN(3,I1,I2)
      CX(3,3)=GBASE(3,1)*SN(1,I1,I2)+GBASE(3,2)*SN(2,I1,I2)
     * +GBASE(3,3)*SN(3,I1,I2)
      PARSQZ=PARSQ*Z
      HM(1,1)=HM(1,1)+PARSQZ*(TAU(1,1)*CX(1,1)+TAU(1,2)*CX(1,2)
     * +TAU(1,3)*CX(1,3))
      HM(1,2)=HM(1,2)+PARSQZ*(TAU(1,1)*CX(2,1)+TAU(1,2)*CX(2,2)
     * +TAU(1,3)*CX(2,3))
      HM(2,1)=HM(2,1)+PARSQZ*(TAU(2,1)*CX(1,1)+TAU(2,2)*CX(1,2)
     * +TAU(2,3)*CX(1,3))
      HM(2,2)=HM(2,2)+PARSQZ*(TAU(2,1)*CX(2,1)+TAU(2,2)*CX(2,2)
     * +TAU(2,3)*CX(2,3))
C
      STRESP(1,I1,I2)=STRESP(1,I1,I2)+PARSQ*(TAU(3,1)*CX(1,1)
     * +TAU(3,2)*CX(1,2)+TAU(3,3)*CX(1,3))
      STRESP(2,I1,I2)=STRESP(2,I1,I2)+PARSQ*(TAU(3,1)*CX(2,1)
     * +TAU(3,2)*CX(2,2)+TAU(3,3)*CX(2,3))
      STRESQ(1,I1,I2)=STRESQ(1,I1,I2)+PARSQZ*(TAU(1,1)*CX(3,1)
     * +TAU(1,2)*CX(3,2)+TAU(1,3)*CX(3,3))
      STRESQ(2,I1,I2)=STRESQ(2,I1,I2)+PARSQZ*(TAU(2,1)*CX(3,1)
     * +TAU(2,2)*CX(3,2)+TAU(2,3)*CX(3,3))
 2059 CONTINUE
      DGM33=DGM33+D33S*PAR
      SUMG=SUMG+GG(3,3)*PAR
      IF(ICOUNT .EQ. 1) CALL ENERGY(3,PAR)
      SGAM33=SGAM33+DGAM33*PAR
      IF(INORML.NE.2) GO TO 169
      TAUSUM=DTM(1,1)+DTM(2,2)+DTM(3,3)
```

```
      DGAMA3=DGAMA3+((((1.-2.*HNU)/EE)*TAUSUM
     * -DGAMMX(1,1)-DGAMMX(2,2)+3.*ALPHA*DTE*P)*PAR
  169 CONTINUE
      DGM33=DGM33/THIC
      AVEG33=SUMG/THIC
      AGAM33=SGAM33/THIC
      IF(ICOUNT .EQ. 1)GOTO 180
      DGDG=2.0*(DGM33-AGAM33)/AVEG33
      DELBAR(I1,I2)=-GBTN+SQRT(GBTN**2-DGDG)
  180 CONTINUE
C               SURFACE STRAINS FOR PLOTTING
      IF(ICOUNT .EQ. 0)GOTO 190
      IF(I1.EQ.IS1 .AND. I2.EQ.IS2) PGAM33=PGAM33+AGAM33
      IF(ICOUNT .EQ. 1) CALL AUXIL(10)
      EPSL1(I1,I2)=EPSL1(I1,I2)+0.5*(DA(1,1)+THIC*DB(1,1))
      EPSL2(I1,I2)=EPSL2(I1,I2)+0.5*(DA(2,2)+THIC*DB(2,2))
      GAMMAL(I1,I2)=GAMMAL(I1,I2)+0.5*(DA(1,2)+THIC*DB(1,2))
      EPSU1(I1,I2)=EPSU1(I1,I2)+0.5*(DA(1,1)-THIC*DB(1,1))
      EPSU2(I1,I2)=EPSU2(I1,I2)+0.5*(DA(2,2)-THIC*DB(2,2))
      GAMMAU(I1,I2)=GAMMAU(I1,I2)+0.5*(DA(1,2)-THIC*DB(1,2))
  190 CONTINUE
      IF(INORML.NE.2) GO TO 200
C     AVERAGING AT EACH POINT
      DGAMA3=DGAMA3/THICKZ
C     SETTING Y2 AND D2
      IF(CIRCH)GO TO 450
      D2(1,I1,I2)=D2(1,I1,I2)+DGAMA3*SN(1,I1,I2)
      D2(2,I1,I2)=D2(2,I1,I2)+DGAMA3*SN(2,I1,I2)
      D2(3,I1,I2)=D2(3,I1,I2)+DGAMA3*SN(3,I1,I2)
  450 Y2(1,I1,I2)=Y2(1,I1,I2)+DGAMA3*SN(1,I1,I2)
      Y2(2,I1,I2)=Y2(2,I1,I2)+DGAMA3*SN(2,I1,I2)
      Y2(3,I1,I2)=Y2(3,I1,I2)+DGAMA3*SN(3,I1,I2)
  200 HM1(1,I1,I2)=HM(1,1)
      HM1(2,I1,I2)=HM(1,2)
      HM2(1,I1,I2)=HM(2,1)
      HM2(2,I1,I2)=HM(2,2)
C
C     ASSUMING ORTHOGONAL COORDINATES, TO REDUCE ERROR
C
      IF(K1.NE.7.AND.K2.NE.7) RETURN
      HM1(2,I1,I2)=0.
      HM2(1,I1,I2)=0.
      RETURN
      END
```

DISTRIBUTION LIST

No. of Copies	Organization
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314
1	Director Defense Advanced Rsch Projects Agency 1400 Wilson Boulevard Arlington, VA 22209
1	Office of Deputy Under-Sec of Defense for Res & Eng (ET) ATTN; Mr. J. Persh Staff Spec for Materi- als & Structures Room 3D1089, The Pentagon Washington, DC 20301
2	Fed Emergency Management Agency ATTN: Mr. M. Pachuta RF-SR, Tech Lib Washington, DC 20472
6	Director Defense Nuclear Agency ATTN: STSI/Archives DDST/Dr. Oswald SPAS,SPSS,SPTD, RATN Washington, DC 20305
1	HQDA (DAMA-AR-A) Washington, DC 20310
1	HQDA (DAMA-WSM) Washington, DC 20310
1	HQDA (DACS-BMZ-A) Washington, DC 20310
1	Commander US Army Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, MD 20014
1	Deputy Chief of Staff for Intelligence USA-Europe ATTN: Tech Sec APO, NY 09131
1	Commander in Chief US Army Europe & Seventh Army ATTN: AEURRIC APO, NY 09403
1	Commandant US Army Command and Gen Staff College ATTN: Archives Ft Leavenworth, KS 66027
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander Armament R&D Center US Army AMCCOM ATTN: DRSMC-TDC(D) Dr. D. Gyorog Dover, NJ 07801
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCCP 5001 Eisenhower Avenue Alexandria, VA 22333

DISTRIBUTION LIST

No. of Copies	Organization
2	Commander Armament R&D Center USA AMCCOM ATTN: DRSMC-TSS(D) Dover, NJ 07801
1	Commander US Army Armament, Munitions and Chemical Command ATTN: DRSMC-LEP-L(R) Rock Island, IL 61299
3	Commander US Army Armament, Munitions and Chemical Command ATTN: DRSMC-LEP-L(R) DRCPM-ARGADS(R) DRSMC-RDF(R) Rock Island, IL 61299
1	Director Benet Weapons Laboratory Armament R&D Center ATTN: DRSMC-LCB-TL(D) Watervliet, NY 12189
1	Director Institute for Defense Analyses ATTN: Library 1801 Beauregard Street Alexandria, VA 22311
2	Dep Chief of Staff for Operations & Plans ATTN: Technical Lib Dir, Chem & Nuclear Operations Department of the Army Washington, DC 20310
1	Commander US Army Missile Command ATTN: DRSMI-R Redstone Arsenal, AL 35898
2	Director US Army BMD Advanced Technology Center ATTN: CRDABH-X, CRDABH-S P.O. Box 1500 Huntsville, AL 35807
1	Commander US Army BMD Systems Command ATTN: BMDSC-TFN, N.J. Hurst P.O. Box 1500 Huntsville, AL 35807
2	Chief of Engineers ATTN: DAEN-MCE-D DAEN-RDM Washington, DC 20314
3	Commander US Army Aviation Research and Development Command ATTN: DRDAV-E DRCPM-ASE DRDAV-DI 4300 Goodfellow Boulevard St. Louis, MO 63120
1	Director US Army Air Mobility Research & Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMA-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Missile Command ATTN: DRSMI-YDL Redstone Arsenal, AL 35898

DISTRIBUTION LIST

No. of Copies	Organization
4	Director Applied Technology Lab US Army Research & Technology Laboratory (AVRADCOM) ATTN: DAVDL-EU-VAT, Mr. Merritt DAVDL-PP, Mr. Morrow DAVDL-ATL DAVDL-RM Fort Eustis, VA 23604
1	Commander US Army Troop Support & Aviation Materiel Readiness Command ATTN: DRSTG-G Sys Anal Ofc 4300 Goodfellow Boulevard St. Louis, MO 63120
2	Commander US Army Communications Research and Development Command ATTN: DRSEL-ATDD DRDCO-PPA-SA Fort Monmouth, NJ 07703
1	Commander US Army Engineer Waterways Exp Station ATTN: Tech Lib P.O. Box 631 Vicksburg, MS 39181
1	Commander US Army Eng Center Fort Belvoir, VA 22060
1	Commander US Army Missile Command ATTN: DRSMI-CM Redstone Arsenal, AL 35898
1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
2	Commander US Army Harry Diamond Lab ATTN: DELHD-TA-L DELHD-RBA/J. Rosado 2800 Powder Mill Road Adelphi, MD 20783
3	Commander US Army Harry Diamond Lab ATTN: Mr. J. Gaul Mr. J. Gwaltney Mr. R. J. Bostak 2800 Powder Mill Rd Adelphi, MD 20783
1	Commander US Army Missile Command ATTN: DRCPM-MDET-PA Redstone Arsenal, AL 35898
1	Commander US Army Mobility Equipment R&D Cmd ATTN: DRDME-WC Fort Belvoir, VA 22060
1	Commander US Army Research Office P.O. Box 12211 Research Triangle Pk, NC 27709

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Nuclear & Chem Agcy ATTN: Tech Lib 7500 Backlick Rd, Bldg 2073 Springfield, VA 22150
1	Commander US Army Natick Research and Development Lab ATTN: DRDNA, Dr. Seiling Natick, MA 01760
1	Commander US Army Tank Automotive Command ATTN: DRSTA-TSL Warren, MI 48090
2	Commander US Army Tank Automotive Command ATTN: DRSTA-UL DRSTA-RWL Warren, MI 48090
1	President US Army Aviation Test Bd Fort Rucker, AL 36360
1	Commander US Army Logistics Management Center Fort Lee, VA 23801
4	Commander US Army Materials and Mechanics Research Ctr ATTN: DRXMR-RD Tech Lib (J. Mescall) (R. Shea) (S.C. Chou) Watertown, MA 02172
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL White Sands Missile Range NM 88002
1	Commander US Army Training & Doctrine Command ATTN: DCST-TDD Fort Monroe, VA 23651
1	Commander US Army Combat Development Experimentation Command ATTN: CSCG Ln Ofc Ford Ord, CA 93941
1	Commander US Army Forces Command ATTN: AVN SEC Fort McPherson, GA 30330
1	Commandant US Army Air Defense Center ATTN: Air Defense Agency Fort Bliss, TX 79916
1	Commandant US Army Armor School ATTN: Armor Agency Fort Knox, KY 40121
1	Commandant US Army Aviation School Fort Rucker, AL 36360
1	Commandant US Army Field Artillery Sch ATTN: Field Artillery Agency Fort Sill, OK 73503

DISTRIBUTION LIST

No. of Copies	Organization
2	Commandant US Army Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905
1	Commandant US Army Transportation School Fort Eustis, VA 23604
1	Commander US Army Combined Arms Combat Dev Activity Ft Leavenworth, KS 66027
1	Commander US Army Air Defense Human Research Unit ATTN: ATHRD Fort Bliss, TX 79916
1	Commander US Army Agency for Aviation Safety Fort Rucker, AL 36360
3	Chief of Naval Operations ATTN: NOP-05 NOP-34 NOP-96C2 Department of the Navy Washington, DC 20350
1	Chief of Naval Material ATTN: MAT-03D15 Dept of Navy Washington, DC 20350
1	Commander US Army Missile Command ATTN: DRCPM-HA Redstone Arsenal, AL 35898
10	Commander Naval Air Systems Command ATTN: Air-5204 (2 cys) Air-605 (3 cys) Air 530 Air-095A Air-53603B Air-330 Air-350 Washington, DC 20360
3	Commander Naval Sea Systems Command ATTN: SEA-62R, Mr. Sieve SEA-62Y SEA-9961 Washington, DC 20360
1	Commander Naval Air Development Ctr, Johnsville ATTN: SAED, Code 5033 Warminster, PA 18974
4	Commander Naval Surface Weapons Ctr ATTN: Code DG-13 Code G-10 Lib Br, DX-21 Dr. W. Soper Dahlgren, VA 22448
5	Commander Naval Surface Weapons Ctr ATTN: Code 040, Code N- Code 433, Code N- Code WX21/Tech Lib Silver Spring, Md 20910
1	Commander US Army Missile Command ATTN: DRCPM-SHO-E Redstone Arsenal, AL 35898

DISTRIBUTION LIST

No. of Copies	Organization
8	Commander Naval Weapons Center ATTN: Code 338, Code 3831, Code 3835, Code 39104 Code 3431, Tech Lib Code 31804, M. Keith Code 533, Tech Lib Code 6031, Dr. Stronge China Lake, CA 93555
1	Commander David W. Taylor Naval Ship R&D Center ATTN: Lib Div, Code 522 Bethesda, MD 20084
1	Commander Naval Research Lab ATTN: CODE 2027, Tech Lib Washington, DC 20375
6	Officer-in-Charge (Code L31) Civil Engineering Lab Naval Construction Btn Ctr ATTN: Dr. W. A. Shaw John Crawford Technical Library Stan Takahashi R. J. Odello W. A. Keenan Port Hueneme, CA 93041
1	Office of Naval Research ATTN: N. Perrone 800 N. Quincy Street Arlington, VA 22217
1	Superintendent Naval Postgraduate School ATTN: Code 57, Prof Ball Monterey, CA 93940

No. of Copies	Organization
1	Commander Naval Weapons Eval Facility ATTN: Doc Control Kirtland AFB, NM 87117
1	Commander Naval Ammunition Depot ATTN: RD-3 Crane, IN 47522
2	Commander Naval Research Lab ATTN: Code 6003 Code 8443 Washington, DC 20375
5	Commandant US Marine Corps ATTN: Code AAP Code AAW-1 Code AAW-5 Code APW-52 Code APW-71 Washington, DC 20380
1	Director Development Center, MCDEC Quantico, VA 22134
4	AFWL/SUL AFWL/PG AFSWC/TEEM AFSWC/TESPO Kirtland AFB, NM 87117
1	HQ USAF (AFSCAGF)/SDOA Andrews AFB, MD 20334
3	HQ AFSC (SDZ, SDDE, DLAW)/SDOA Andrews AFB, MD 20334

DISTRIBUTION LIST

No. of Copies	Organization
3	US Army Field Office /SDOA HQ USAF Systems Command ATTN: SDNE DDAB, SDW Andrews AFB, MD 20334
1	ADTC (SES) Eglin AFB, FL 32542
2	AFATL (DRLRV, J. R. Rutland) (DLYV) Eglin AFB, FL 32542
1	USAFTAWC (OA) Eglin AFB, FL 32542
1	TAC Langley AFB, VA 23365
1	AFAPL (SFH) Wright-Patterson AFB, OH 45433
1	AFFDL (FES) Wright-Patterson AFB, OH 45433
1	AFFDL (FER) Wright-Patterson AFB, OH 45433
1	AFFDL (FEN) Wright-Patterson AFB, OH 45433
1	AFFDL (FDTR, Dr. Janik) Wright-Patterson AFB, OH 45433
1	ASD/XROT, G. B. Bennett Wright-Patterson AFB, OH 45433
1	AFML (LLN, Dr. Nicholas) Wright-Patterson AFB, OH 45433
1	ASD/ENFTV, Mr. J. Wallick Wright-Patterson AFB, OH 45433

No. of Copies	Organization
1	Director Lawrence Livermore Lab Tech Info Div P.O. Box 808 Livermore, CA 94550
2	Director Los Alamos Scientific Lab ATTN: Doc Control P.O. Box 1663 Los Alamos, NM 87544
1	Director NASA Scientific and Tech Info Facility P.O. Box 8757 BaltWash Intern'l Airport, MD 21240
1	AFFDL (FBE, Dr. Bader) Wright-Patterson AFB, OH 45433
2	Director National Aeronautics and Space Administration Langley Research Center ATTN: MS 246E, L. Bement MS 249, J. Ward Hampton, VA 23665
1	Director National Aeronautics & Space Administration Lewis Research Center 2100 Brookpark Road, MS 77-5 Cleveland, OH 44135
1	Bell Helicopter Textron ATTN: Mr. Nile Fischer P.O. Box 482 Fort Worth, TX 76101

DISTRIBUTION LIST

No. of Copies	Organization
1	The Boeing Company Vertol Division ATTN: Dave Harding P.O. Box 16858 Philadelphia, PA 19142
1	Fairchild Industries Fairchild Republic Div ATTN: D. Watson Farmingdale, LI, NY 11735
1	KETRON Company Thor Division 696 Fairmount Avenue Baltimore, MD 21204
2	Falcon Research and Development Company ATTN: Levelle Mahood Arthur Stein 109 Inverness Drive, E Englewood, CO 80112
1	General Electric Company Military Engine Products Division ATTN: E. L.. Richardson Building 2406A 1000 Western Avenue West Lynn, MA 01905
1	Kaman Tempo ATTN: E. Bryant 715 Shamrock Rd Suite UL-1 Bel Air, MD 21014
1	Hughes Aircraft Co. ATTN: Security Officer Centinela and Teale Sts Bldg 6, MS E-125 Culver City, CA 90230
2	Director Sandia National Labs ATTN: Info Dist Div Dr. W. Von Riesmann Albuquerque, NM 87115
2	University of Delaware Dept of Mech & Aerospace Engineering ATTN: Prof. J. R. Vinson Prof. M. Taya Newark, DE 19711
1	Florida Atlantic Univ Dept of Ocean Eng ATTN: Prof. K. Stevens Boca Raton, FL 33432
1	J.G. Eng Res Associates 3831 Menlo Drive Baltimore, MD 21215
1	Mass Inst Of Tech Aerolastic & Structures Research Laboratory ATTN: Dr. E. A. Witmer 77 Massachusetts Avenue Cambridge, MA 02139
1	ASD (XRM) Wright-Patterson AFB, OH 45433
1	Texas A&M University Dept Aerospace Eng ATTN: Dr. W. Haisler College Station, TX 77843
1	Weidlinger Associates Consulting Eng ATTN: M. L. Baron 110 East 59th Street New York, NY 10022
2	Virginia Polytechnic Inst & State Univ Dept of Eng Mech ATTN: Prof D. Frederick Prof C. W. Smith Blacksburg, VA 24061

DISTRIBUTION LIST

No. of Copies	Organization
1	Georgia Inst of Tech School of Civil Eng ATTN: Prof S. N. Atluri Atlanta, GA 30332
1	Brown University Div of Eng ATTN: Prof. P. Symonds Providence, RI 02912
1	University of Dayton Res Inst ATTN: Dr. S. Bless Dayton, OH 45469
1	University of Washington Dept of Aeronautics and Astronautics ATTN: Prof. I. M. Fyfe Seattle, WA 98105
1	LTV Aerospace Corp Vought Systems Division ATTN: D. M. Reedy Unit 2 54244 P.O. Box 5907 Dallas, TX 75222
1	Meredith Engineering 33170 Glen Valley Drive Farmington Hills, MI 48018
1	Aeronautical Research Associates of Princeton, Inc. ATTN: Dr. J. W. Leech P.O. Box 2229 50 Washington Road Princeton, NJ 08540
1	Rockwell International Los Angeles Aircraft Div ATTN: R. L. Moonan Mail Code AB75 International Airport Los Angeles, CA 90009

No. of Copies	Organization
1	United Aircraft Corp Sikorsky Aircraft Div ATTN: S. O. Karma Stratford, CT 06602
1	New Mexico Inst of Minning and Technology TERA Group Socorro, NM 87801
1	Southwest Research Inst ATTN: W. Baker 8500 Culebra Road San Antonio, TX 78228

ABERDEEN PROVING GROUND

Dir, USAMSAA
ATTN: DRXSY-D
DRXSY-MP, H. Cohen
Cdr, USATECOM
ATTN: DRSTE-TO-F
Cdr, CRDC, AMCCOM
ATTN: DRSMC-ACW
DRSMC-CLB-PA
DRSMC-CLN
DRSMC-CLJ-L

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number ______________________

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

____

____

____

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) ______________________

____

____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

____

____

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) ______________________

____

____

____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: ______________________

Telephone Number: ______________________

Organization Address: ______________________

____

____

— — — — — — — — — — FOLD HERE — — — — — — — — — —

Director
US Army Ballistic Research Laboratory
ATTN: DRSMC-BLA-S (A)
Aberdeen Proving Ground, MD 21005

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC

POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
US Army Ballistic Research Laboratory
ATTN: DRSMC-BLA-S (A)
Aberdeen Proving Ground, MD 21005

— — — — — — — — — — FOLD HERE — — — — — — — — — —

END

FILMED

DTIC